

**NASA CONTRACTOR  
REPORT**

**NASA CR-2885**



**NASA CR**

0061717



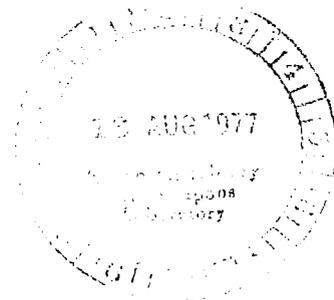
TECH LIBRARY KAFB, NM

**LOAN COPY: RETURN TO  
AFWL TECHNICAL LIBRARY  
KIRTLAND AFB, N. M.**

**COMPUTERS FOR REAL TIME  
FLIGHT SIMULATION:  
A MARKET SURVEY**

*George A. Bekey and Walter J. Karplus*

*Prepared by*  
**COMPUTER SCIENCES CORPORATION**  
Mountain View, Calif. 94043  
*for Ames Research Center*





0061717

1. Report No. NASA CR-2885	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle "Computers for Real Time Flight Simulation: A Market Survey"		5. Report Date July 1977	6. Performing Organization Code
		8. Performing Organization Report No.	
7. Author(s) George A. Bekey and Walter J. Karplus		10. Work Unit No.	
9. Performing Organization Name and Address Computer Sciences Corporation 1101 San Antonio Road Mountain View, California 94043		11. Contract or Grant No. NAS 2-7806	
		13. Type of Report and Period Covered Contractor Report	
12. Sponsoring Agency Name and Address National Aeronautics & Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code	
15. Supplementary Notes			
16. Abstract  An extensive computer market survey to determine those available systems suitable for current and future flight simulation studies at Ames Research Center. The primary requirement is for the computation of relatively high frequency content (5 Hz) math models representing powered lift flight vehicles. The Rotor Systems Research Aircraft (RSRA) was used as a benchmark vehicle for computation comparison studies. The general nature of helicopter simulations and a description of the benchmark model are presented, and some of the sources of simulation difficulties are examined. A description of various applicable computer architectures is presented, along with detailed discussions of leading candidate systems and comparisons between them.			
17. Key Words (Suggested by Author(s))  Flight Simulation		18. Distribution Statement  UNCLASSIFIED-UNLIMITED  STAR Category 05	
19. Security Classif. (of this report) UNCLASSIFIED	20. Security Classif. (of this page) UNCLASSIFIED	21. No. of Pages 82	22. Price* \$4.00

## TABLE OF CONTENTS

I.	INTRODUCTION AND OVERVIEW OF THE SURVEY	1.1
1.1	Background	1.1
1.2	General Requirements	1.1
1.3	Method of Approach	1.2
1.4	General Philosophy of Approach	1.2
1.5	Organization of the Report	1.3
II.	MATHEMATICAL MODELS OF ROTARY WING AIRCRAFT	2.1
2.1	The Nature of the Helicopter Modeling Problem	2.1
2.2	Major Elements of the Mathematical Model	2.2
2.3	Rotor Model	2.7
2.4	Computational Solution of the Rotor Model	2.12
2.5	Summary of the Simulation Difficulties	2.13
III.	THE BENCHMARK PROBLEM	3.1
3.1	Summary of the Mathematical Model	3.1
3.2	Digital Operations Involved in the Benchmark Problem	3.7
3.3	Simulation Frame Time Requirements	3.8
3.4	Limitations and Extensions of the Benchmark Problem	3.9
3.5	Summary	3.10

IV.	ALTERNATIVE ARCHITECTURAL APPROACHES	4.1
4.1	General Purpose vs. Special Purpose Processors	4.1
4.2	Array Processors	4.3
4.3	Simulation-Oriented Peripheral Processors	4.5
4.4	Arrays of Processors	4.5
4.5	Hybrid Computers	4.7
4.6	Other Possibilities	4.9
V.	DESCRIPTION OF LEADING CANDIDATES	5.1
5.1	Floating Point Systems, Inc. - AP-120B	5.2
5.2	Applied Dynamics, Inc. - AD-10	5.14
5.3	Denelcor, Inc. - HEP	5.25
5.4	W.W.Gaertner Research, Inc. -G-471	5.31
5.5	Paragon Pacific, Inc. - SPURS	5.37
VI.	COMPARISONS AND TRADEOFFS	6.1
6.1	Floating Point Systems AP-120B vs. Applied Dynamics AD-10	6.3
6.2	AP-120B or AD-10 vs. Denelcor HEP or Gaertner G-471	6.6
6.3	Paragon Pacific SPURS vs. Digital Processors	6.8
VII.	CONCLUSIONS	7.1
VIII.	RECOMMENDATIONS	8.1
	APPENDIX	A.1
	List of Organizations and Individuals Contacted	

## I INTRODUCTION AND OVERVIEW OF THE SURVEY

### 1.1 Background

In October, 1976, Computer Sciences Corporation entered into an agreement with the authors of this report to conduct an extensive computer market survey to determine those computer systems which would be most suitable for current and future flight simulation studies at the NASA/Ames Research Center. The primary motivation for the survey arose from the inadequacy of the computers presently used for flight simulation at Ames, particularly with respect to powered-lift vehicles. These vehicles, including rotorcraft, powered lift STOL vehicles, and lift-fan V/STOL vehicles, exhibit dynamic response modes in the range of 30-100 radians per second. Attempts to simulate such vehicles with the Sigma family of computers currently available at Ames result in frame times which are excessively long. Hence, it became evident that the requirement for real time simulation of this class of vehicles was impossible to meet with presently available computers without extreme simplifications of the mathematical models.

### 1.2 General Requirements

In addition to the general objective of recommending a computer system suitable for the simulation of rotary wing aircraft and powered lift vehicles, the authors were charged with the selection of a system compatible with the distributed nature of the overall computation and simulation system at NASA/Ames. It became clear early in the study that a super-computer to be time shared among a number of simulation facilities

was not compatible with this overall concept. Rather, the distributed network (known as the Olympus Concept) was to feature a moderate sized computer as a host machine, connected by extensive digital communication lines to dedicated computers associated with specific fixed and moving base simulators, picture generation equipment and display generation facilities.

An additional motivation for this study arises from the fact that a large vertical moving base simulator, specifically designed for the simulation of the class of vehicles under consideration is under construction at the present time.

### 1.3 Method of Approach

The approach to the problem was as follows:

- A. A mathematical model to be used as a benchmark for evaluation of candidate systems was defined.
- B. Prior experiences in helicopter simulation, both real time and non-real time were reviewed and evaluated. This task was accomplished by a combination of technical discussions with appropriate individuals and a survey of available literature.
- C. Numerous conferences were held with individuals in the computer field at universities, government agencies and private industry, to ascertain the state of the art of both currently-available and projected computer equipment.
- D. A number of visits to specific vendors was conducted in order to obtain detailed information concerning those systems and peripherals which appeared most suitable for the task.

A complete list of individuals and organizations contacted in the course of the computer survey is included as an Appendix to this report.

#### 1.4 General Philosophy of Approach

In order to accomplish the computer survey within the allocated time frame, two restrictions were placed on the investigation; first, a specific simulation was selected as a "benchmark" for the evaluation of prospective computer hardware and second, detailed consideration of equipment was restricted to those manufacturers who had operational hardware currently in the field or prototype equipment in a sufficiently advanced stage of development so that delivery of both hardware and software within the next twelve months could prove to be a reasonable possibility. Furthermore, since the survey was intended to find an economical computer system that will meet the needs of flight simulation during the next few years, the survey was restricted to systems with hardware costs in the range of \$100,000-\$750,000, thus omitting from serious consideration super-computers with prices in the multi-million dollar range.

The two types of aircraft that will demand the most computational power in the next few years are the lift fan vehicles and the helicopters. The first step was to determine which of these type vehicles was the most demanding on the computer and choose it as the benchmark simulation. Numerous consultations with knowledgeable individuals both at NASA/Ames and elsewhere led to the conviction that a simulation system capable of representing a helicopter type vehicle with dynamic response modes in the range of 30-100 radians per second would also be capable of representing STOL or lift-fan vehicles. This led to the selection of the Rotor Systems Research Aircraft (RSRA), operating in its helicopter mode, as a benchmark vehicle.

Following the selection of the RSRA, the history of the mathematical models being used for its simulation at Sikorsky Aircraft Company and at the NASA/Langley Research Center was investigated, with particular attention being paid to the work of J. Houck who used the same vehicle in an attempt to investigate tradeoffs between accuracy, frame time, and complexity of mathematical representation of the rotor. The FORTRAN program being used for the current simulation of this vehicle at NASA/Ames was reviewed in detail and analyzed in order to determine the number of mathematical operations (e.g., multiplications, function generations, additions, etc.) required for its implementation. The operation count arising from this analysis was then used to compute estimated timings for the solution of the model equations on a variety of candidate systems.

#### 1.5 Organization of the Report

The remainder of this report is divided into seven chapters. Chapter 2 is specifically concerned with the mathematical model of rotary wing aircraft and its implementation in both real time and non real time. The general nature of the helicopter simulation problem is analyzed and some of the specific sources of difficulty are examined. Chapter 3 considers the specific benchmark helicopter model and offers some comments regarding its limitations and the possible need for expansion of future models. The operation count involved in the representation of the benchmark problem is presented in detail.

Chapter 4 presents a general discussion of alternative approaches to computer architectures which are compatible with the rotorcraft flight simulation problem. These include so-called array processors which make use of parallelism and pipelining, architectural features which result in extremely high internal computational speeds particularly if applied

to problems in which large arrays of numbers undergo generally similar mathematical operations. Other approaches considered include processors which are especially designed for dynamic system simulation, parallel arrays of processors, hybrid computers, and digital differential analyzers.

Chapter 5 provides a detailed discussion of the leading candidate systems for the helicopter flight simulation task. The specific advantages and disadvantages of each system are presented in considerable detail. Chapter 6 is a discussion of the trade-offs between the candidate systems, on the basis of such criteria as speed, programmability, software availability, proven performance, cost and versatility. Conclusions are summarized in Chapter 7. The final chapter presents recommendations both for specific types of equipment and for a schedule of tasks and acquisitions to be followed in the transition from the present equipment to a new flight simulation facility.

## II. MATHEMATICAL MODELS OF ROTARY WING AIRCRAFT

### 2.1 The Nature of the Helicopter Modeling Problem

The development of mathematical representations of the dynamics of a helicopter and their computational solution has represented a major challenge to engineers and computer scientists for many years. In contrast with fixed wing aircraft, a helicopter is characterized by a rotating lifting surface so that the primary component of air velocity at each rotor blade is due to blade rotation and not to aircraft velocity. Hence, as may be seen below, aerodynamic forces depend on the radial distance from the hub. In addition these forces depend on the blade azimuth. Furthermore, the rotating blades of wings are sufficiently elastic so that bending or aeroelastic behavior may have to be included in a comprehensive mathematical model.

In addition to problems that arise from representation of the rotor, which we shall consider in greater detail, the helicopter is characterized by a variety of flight regimes including vertical takeoff, hover and transitional modes which also present significant problems in modeling. In particular, interference effects arising from downwash from the moving blades onto the fuselage, or ground interference effects which occur during landing or hover near the ground, are imperfectly understood and hence are usually characterized by empirical models with varying degrees of success.

In recent years the situation has been even more complicated by the development of novel types of rotary wing aircraft. For example, the rotor systems research aircraft (RSRA) is a vehicle capable of

takeoff and landing as a helicopter but forward flight as a fixed wing aircraft. A number of novel concepts for helicopter rotors, such as the Sikorsky ABC (Advancing Blade Concept) which consists of 2 three-bladed coaxial rotating systems and present new challenges in representation have been developed.

The net effect of this complexity has been the development of a large gap between mathematical models being used for engineering development of rotorcraft and those models being used for real time simulation or real time, man-in-the-loop simulation. The major engineering programs run on large digital computers of the IBM 370 or CDC 6600 class at speeds ranging from .01 - .05 of real time. Clearly, any attempt to use such programs for real time simulation requires drastic simplification of the mathematical models, with consequent losses of model fidelity and restrictions on the types of maneuvers and operational conditions which can be simulated. In the following paragraphs, we shall review briefly the overall mathematical model, and then indicate some of the simplifications and computational problems which arise in attempts to simplify these equations for real time simulations.

## 2.2 Major Elements of the Mathematical Model

During the past ten years a number of manufacturers has developed comprehensive mathematical models of rotorcraft behavior. Among these the best known are the C-81 program (developed at Bell Helicopter Co.), the REXOR Program (developed by Lockheed Aircraft Co.) and the NORMAL MODES program (developed by Sikorsky Aircraft Co.). An indication of the complexity of the mathematical model may be obtained by considering that the C-81 program consists of approximately 30,000 cards requiring 900K of core storage on an IBM 360 computer. In block diagram form the

C-81 program may be represented by eleven major groups of equations as shown in Figure 2.1. The major composition of this block diagram is the following:

Block 1:

This block represents the effect of control signals applied by the pilot in the form of both stick position and pedal position. Since directional control of a helicopter is affected by means of adjustment of rotor blade angles, the primary output of the control system block is to Block 6 which represents the rotor aerodynamics. In addition, stabilizing surfaces on the main fuselage and jet thrust are under pilot control. These equations include a number of nonlinearities since stick and pedal position, jet thrust and control surface position are all characterized by maximum and minimum values.

Block 2:

Consists of the equations which represent the SCAS (Stability and Control Augmentation System) which simulates the hardware which provides improved stability and response to pilot inputs. Specifically, the SCAS provides additional feed forward and feedback elements, which improve the pilot's ability to control the vehicle. The SCAS has independent channels for roll, pitch and yaw. Approximately 30 other transfer functions are used for both forward and feedback elements. Their simulation requires numerical differentiation to obtain the necessary derivatives with appropriate correction equations to reduce the noise generated by this process.

Block 3:

The angle of attack and slip angle are used to compute the aerodynamic coefficients for each stabilizing surface. The definition of

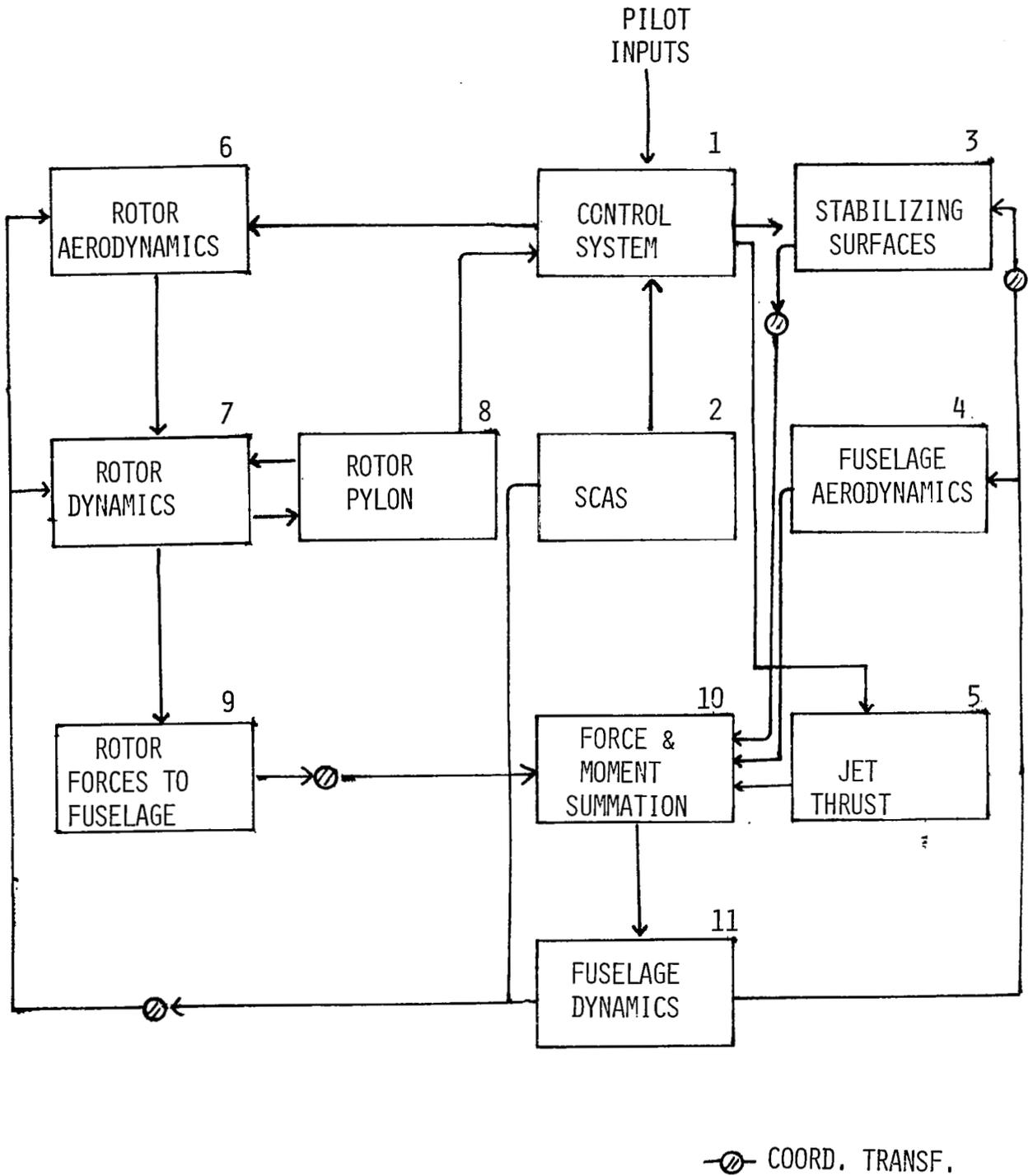


Fig. 2.1 C-81 Block Diagram

these angles requires that the body axis components of the local air-flow velocity be transformed into the local reference system for each surface. This axis transformation is indicated by the shaded circle in the block diagram of Figure 2.1. Then the lift, drag and pitching moment coefficients ( $C_L, C_D, C_M$ ) are computed. The C-81 program provides the user with an option for computing these coefficients from either sets of data tables or from equations. The tables representing the stabilizing surfaces include the 3-dimensional air foil characteristics. A number of other problems complicate this block, such as correction of the aerodynamic coefficients for the aspect ratio of each wing or stabilizer, sweep effects, and so forth.

Block 4:

This block represents the force and moment equations which characterize the helicopter fuselage including lift and drag forces, pitching moment, side force, rolling moment and yawing moment.

Block 5:

Represents the engine dynamics in response to control commands from the pilot.

Block 6:

This block represents the calculation of the aerodynamic forces on the rotor blades which are needed for determining the various components of blade velocity which are computed in Block 7. As indicated in connection with Block 3, both equations and tables (which are functions of angle of attack and Mach number) are available for calculation of the aerodynamic coefficients.

Block 7:

This block represents the calculation of the components of blade velocity as a function of the aerodynamic forces, gravity forces, and forces transmitted through the rotor pylon which in turn depend on the method of attachment of the particular blade. In this block the equations representing various degrees of freedom of the rotor are represented, including blade flapping (out of plane movement) and lagging (in plane oscillation). In addition, aeroelastic effects which give rise to the bending of the blades are represented by means of normal mode equations which are discussed below in Section 2.3.

Block 8:

This block represents the interaction of the rotor dynamic equations with the attachment mechanisms and support structures.

Block 9:

Here the forces and moments transmitted to the fuselage through the hinge or rigid attachment points of each rotor blade are calculated.

Block 10:

The forces and moments which affect the behavior of the fuselage are summed in the equations which comprise this block. The major forces acting on the fuselage are those produced by the rotor, the stabilizing surfaces, the aerodynamics of the fuselage itself, and the thrust produced by the engines. It clearly shows, forces calculated in local coordinates in other blocks must first be transformed to body coordinates for the summation.

Block 11:

This block represents the computation of fuselage velocity components as a function of the forces and moments calculated in Block 10.

### 2.3 Rotor Model

The major difficulty encountered in the simulation of helicopter equations arises from the complexities associated with the rotor model. As we have indicated in Section 2.1 in a fixed wing aircraft the velocity of the air at the lifting surface depends primarily upon the aircraft speed. On the other hand, for a helicopter, the primary component of velocity is due to the rotation of blades and hence varies as a function of blade radius. In forward flight the angle between major components of the air velocity varies with the blade azimuth thus giving a periodic variation of the magnitude of the velocity vector. This results in periodic blade loads so that the response of the blade and the fuselage can be expressed in multiples of the blade angular velocity,  $\Omega$ . Hence, real time simulation requires the ability to simulate frequencies of at least once per revolution of the rotor blades.

While a detailed derivation of the rotor model equations is beyond the scope of this report, a few basic expressions will be given in order to indicate some of the sources of computational difficulty. Consider first the basic blade element aerodynamics by reference to Figure 2.2. We consider here only a simplified set of equations which omits a number of additional complexities present in the C-81 model.

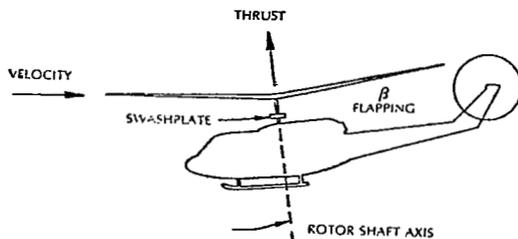


Fig. 2.2 Outline Drawing of Basic Helicopter

In forward flight, a blade encounters an air flow due to the forward velocity of the helicopter which adds to the rotational velocity of the blade as it advances into the direction of flight and subtracts from the rotational velocity of the blade as it retreats. This can be seen in Figure 2.3. The unequal flow causes more lift to be developed on the advancing than the retreating side. If we assume that the blades are hinged (as is true in a number of typical configurations) to allow free vertical or flapping motion, the blade will flap up as it advances and down as it retreats in an attempt to equalize the lift. A downward air flow, the inflow velocity, also acts on the blade. This flow is composed of a component of the helicopter flight velocity due to shaft tilt with respect to the flight and an induced downwash due to the development of thrust.

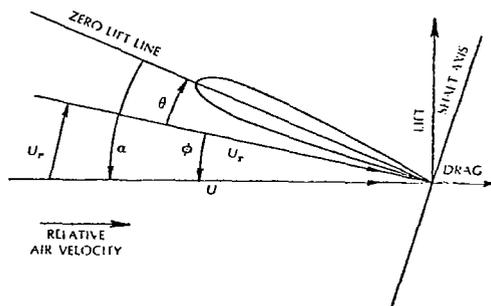


Fig. 2.3 Blade Element in Forward Flight

The basic equations for the vertical and horizontal components of air velocity are :

$$u_t = \Omega r + V \sin \Psi \quad (2.1)$$

$$u_p = W - \dot{\beta}r - \beta V \cos \Psi \quad (2.2)$$

$\Omega$  is the blade rotational velocity,  $r$  is the radial distance from the center of the shaft,  $V$  is the flight speed,  $\Psi$  is the azimuth position of the blade in the plane of rotation,  $W$  is the inflow velocity and  $\beta$  is the flapping angle. Using these velocity components and the blade pitch deflection  $\theta$  it is now possible to calculate the angle of attack from:

$$\alpha = \theta + \tan^{-1}(u_p/u_t) \quad (2.3)$$

Knowing the angle of attack and the total velocity or Mach number, it is now possible to calculate the lift and drag forces and the pitching moment at any particular radial distance from the expressions:

$$L = Q cr C_L \quad (2.4)$$

$$D = Q cr C_D \quad (2.5)$$

$$M = Q c^2 r C_M \quad (2.6)$$

where  $c$  is the blade chord,  $C_L$  is the lift coefficient,  $C_D$  is the drag coefficient,  $C_M$  is the pitching moment coefficient, and  $Q$  is the dynamic pressure defined by:

$$Q = \rho (u_t^2 + u_p^2) / 2 \quad (2.7)$$

where  $\rho$  is the air density. Clearly the forces acting on the blade are thus functions of the total velocity. The lift, drag and pitching moment coefficients are functions of angle of attack and Mach number, and extensive tables of these are available.

Equations of motion representing blade displacement in the vertical and horizontal directions are obtained by considering each blade as a cantilever beam with a distributed dynamic air load which depends on the radial distance and on time,  $t$ . The basic equation is of the form:

$$\frac{\partial^2}{\partial x^2} \left[ EI(x) \frac{\partial}{\partial x^2} y(x,t) \right] + \frac{\partial}{\partial x} \left[ T \frac{\partial y(x,t)}{\partial x} \right] + m(x) \frac{\partial^2}{\partial t^2} y(x,t) = F(x,t) \quad (2.8)$$

where  $y$  is the deflection,  $EI$  is the bending stiffness,  $T$  is the blade tension,  $m$  is the mass distribution and  $F$  is the external force. This fourth order partial differential equation is extremely difficult to solve. It is generally solved by a separation of variables technique, that is by assuming that the deflection may be represented as a product of two functions, one depending only on time and one depending only on the spatial variable,  $x$ , that is:

$$y(x,t) = \phi(x) \delta(t) \quad (2.9)$$

Equation 2.9 can now be substituted into Equation 2.8 and two equations can be obtained, the homogeneous parts of which take the form:

$$\left[ EI(x) \phi''(x) \right]'' + m\omega^2 \phi(x) = 0 \quad (2.10a)$$

$$\ddot{\delta}(t) + \omega^2 \delta(t) = 0 \quad (2.10b)$$

If one now applies initial and boundary conditions one obtains the specific values of  $\omega_i$  (the eigenvalues) for which these equations are valid, and the corresponding eigenfunctions,  $\phi_i(x)$ . The eigenvalues represent the natural frequencies of oscillation and the eigenfunctions represent the shapes of specific modes of vibration. The complete separated equations now take the form:

$$\left[ EI(x) \phi_i''(x) \right]'' + \omega_i^2 m(x) \phi_i(x) = 0 \quad (2.11a)$$

$$\bar{m}_i \ddot{\delta}_i(t) + \bar{m}_i \omega_i^2 \delta_i(t) = \bar{F}_i(t) \quad (2.11b)$$

where again the primes indicate differentiation with respect to space and the dots represent the differentiation with respect to time.

Equation 2.11a is solved in advance for the mode shapes  $\phi_i(x)$  which are then used in the calculation of a generalized mass  $\bar{m}_i$  and the generalized forcing function  $\bar{F}_i$  which are defined by the general equations:

$$\bar{m}_i = \int_0^L m(x) \phi_i^2(x) dx \quad (2.12)$$

$$\bar{F}_i = \int_0^L F(x,t) \phi_i(x) dx \quad (2.13)$$

In turn,  $\bar{m}_i$  and  $\bar{F}_i$  are used to obtain the time function  $\delta_i(t)$  in Equation 2.11b. The total solution for the deflection is then obtained by summing the solutions of enough modes to accurately represent the blade deflection,

$$y(x,t) = \sum_{i=1}^n \delta_i(t) \phi_i(x) \quad (2.14)$$

The external force,  $F(x,t)$  is a combination of inertial and aerodynamic loads, including lift, drag and pitching moment. Modeling this applied force with distributed loads is enormously complex and many simplifications are often employed.

#### 2.4 Computational Solution of the Rotor Model

There are two commonly used approaches to computer implementation of the rotor model, differing primarily in the way the spatial variables are handled. If continuous mode shapes are computed and stored, then analog integration of Equations 2.12 and 2.13 can be used to obtain the generalized mass and forcing functions for solution of Equation 2.11b. On a high speed analog computer it is possible to perform the integrations along this space variable at a high repetitive speed. On the other hand, for digital computer solution, a finite element representation of the blade can yield a matrix equation in which forces and displacement are calculated at only a specific number of points or sections along the blade. As many as 26 finite elements may be needed to get enough accuracy for the representation of the second deflection mode. To represent four bending modes in the vertical, horizontal, and torsional directions, the C-81 program may solve as many as 84 second-order equations, similar in form to Equation 2.11b, for the generalized coordinate  $\delta_i$ .

Simulation of the rotor equations is the major and overwhelming source of difficulty in the complete real time helicopter simulation, consuming of the order of 75% of the total computation time.

## 2.5 Summary of the Simulation Difficulties

The major difficulties as indicated in the simulation of the rotor blades are:

A. The high dimensionality needed to maintain accuracy in the solution of the generalized deflection equation (i.e., the digital solution, the requirements for a sufficiently high number of finite elements to maintain good accuracy).

B. The magnitude of the tables required in the computation of the blade loads. These tables for the lift, drag and moment coefficients,  $C_L$ ,  $C_D$ , and  $C_M$  are all functions of angle of attack and Mach number. When these equations are implemented in the computer, table look-ups must be performed at each station along the blade. For example, if a blade is represented by 10 finite elements, it would be necessary to perform 30 table look-ups for each time frame simply to solve the equation of motion for the blade using only rigid blade flapping (the first mode of deflection). If aeroelastic (bending) modes are included in addition, the complexity of simulation grows very rapidly.

If, on the other hand, the equations are solved using analog/hybrid computers, it may be more desirable to approximate  $C_L$ ,  $C_D$ , and  $C_M$  by equations which are then used in the calculation of the generalized force. Otherwise, digital function generation techniques may be required along with A-D and D-A conversion equipment in order to transmit the appropriate functions to the analog integration equipment.

To the authors' knowledge, real time simulation of helicopter models always involves a significant degree of simplification from models of the complexity embodied in the Bell-C-81 program. A specific simplified model is discussed in the following section.

### III. THE BENCHMARK PROBLEM

This Chapter presents a brief analysis of the specific helicopter simulation selected as the benchmark problem for the survey, including some comments on the simplifications inherent in this model as compared to a more complex C-81 simulation. The benchmark model is then analyzed on the basis of the major digital operations required for its implementation. Simulation frame time requirements are then analyzed for this model in the light of recent experience both at NASA/Ames and NASA/Langley, and with the objective of providing a time step compatible with real time simulation. Finally, the chapter discusses some of the limitations of the benchmark problem and the potential for expansion to problems of greater complexity.

#### 3.1 Summary of the Mathematical Model

The model selected as the benchmark problem is based on a set of equations originally written at Sikorsky Aircraft Company, later implemented at Langley Research Center, and most recently run at Ames Research Center. In block diagram form the model is shown in Figure 3.1. (This block diagram and its analysis has been provided by Professor R. M. Howe of the University of Michigan as a result of an investigation of the model being performed concurrently with the market survey for NASA). The block diagram (and hence the benchmark problem) represents only the rotor equations, since the major computing effort is concerned with analysis of the dynamics and kinematics of the rotor and the calculation of the resulting forces are moments which the rotor

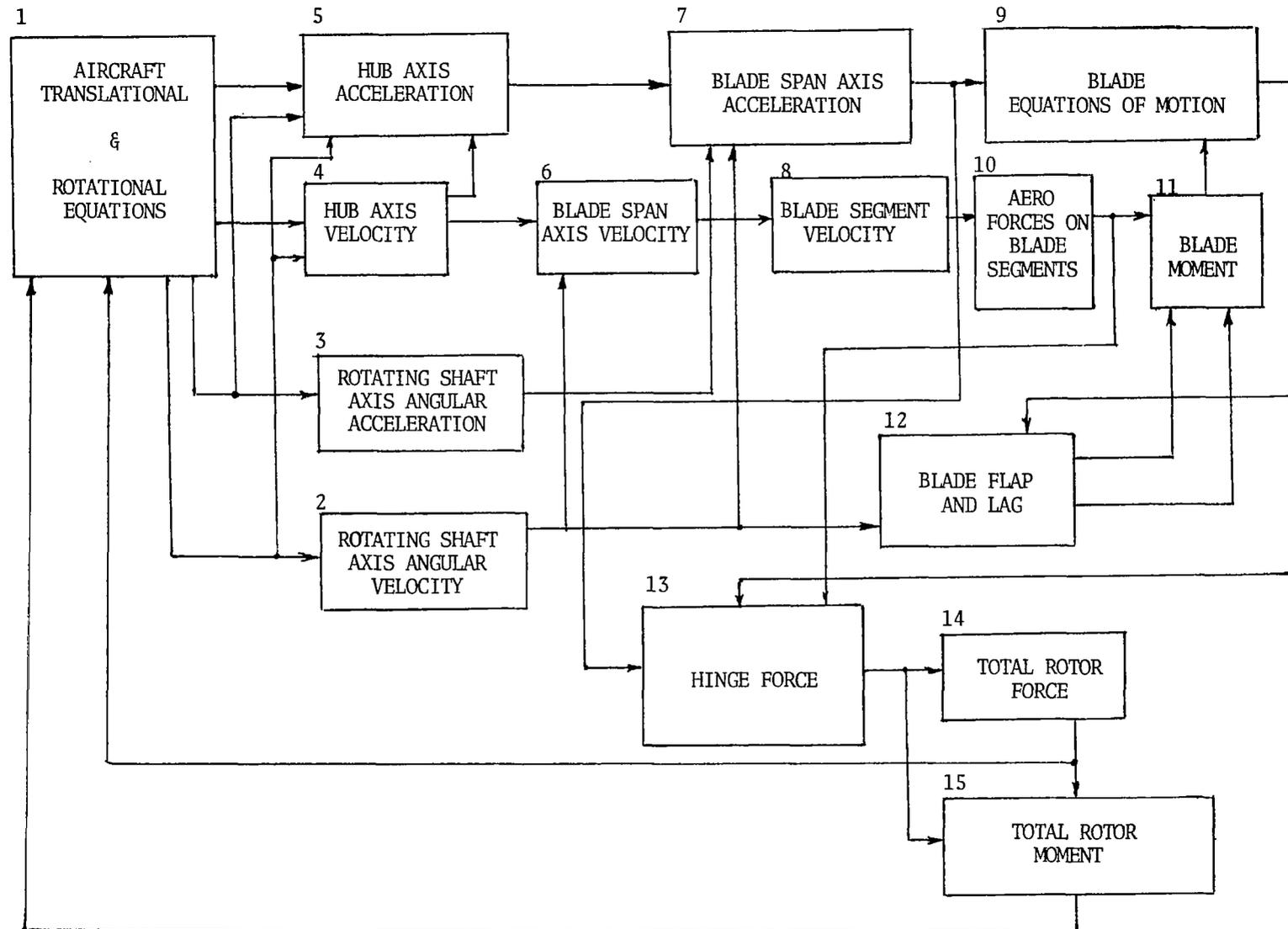


FIG. 3.1 BLOCK DIAGRAM OF HELICOPTER MODEL

imparts on the rigid body.\* We shall analyze each block briefly.

#### Block 1:

This block includes the six degree of freedom equations of motion of the rigid fuselage. The state variables are the translational and rotational velocity components along the body axes, as well as the Euler angles required for transformation to inertial coordinates. Integration of the velocities and coordinate transformation leads to the calculation of distance North, distance East, and altitude. It can be seen that these equations are determined by summing forces and moments about the aircraft center of gravity, but that only the forces and moments produced by the rotor are included explicitly. Other forces, such as those produced by aerodynamic forces on the fuselage itself or its stabilizing surfaces, and the effect of jet engine thrust are not included in this module. The control system and stability and control augmentation models are also omitted.

#### Block 2 and 3: Rotating Shaft Axis Angular Velocity and Acceleration

Blocks 2-3 represent the equations for the accelerations and velocities of the rotor hub axes and rotating rotor shaft axes. Specifically, in block 2, the body axis angular velocities are transformed to rotating shaft axis angular velocities along each blade axis. Since there is a rotating shaft axis system for each blade, the basic transformation equations are repeated N times for N blades, with each blade having a different azimuth angle. Numerous trigonometric quantities must be computed in this block.

---

\*Simulation of the omitted portions of the model (fuselage aerodynamics, engine and stability and control system) represents less than 25% of the computing effort.

In Block 3, the angular acceleration of the rotating shaft axes is computed by using angular accelerations of the body axes and appropriate coordinate transformations. Here again, the basic accelerations must be computed separately for each blade.

#### Block 4 and 5: Hub Axis Velocity and Acceleration

Similarly, the hub axis velocity is computed by making use of the velocity of the body axes (i.e., the aircraft center of gravity and the velocity with respect to the aircraft center of gravity.)

Block 5 equations are used to compute the hub axis and thus the rotating shaft axis acceleration components along the shaft axes. Components of the accelerations of gravity with respect to these axes are included.

#### Block 6 and 7: Blade Span Axis Velocity and Acceleration

Block 6 represents the equations which are used to compute the rotating shaft axis components of the hinge velocity. Since there are N blades, all equations must be solved N times. This block primarily represents a coordinate transformation and the appropriate position of each hinge point with respect to the hub.

Similarly, Block 7 represents the acceleration of the hinge point with respect to the inertial reference frame, with appropriate corrections for gravity.

#### Block 8: Blade Segment Velocity

This block contains the equations for calculating the velocities of the individual blade segments which are needed in order to compute the aerodynamic forces acting on each of the blade segments. If each blade is divided into s segments, it is necessary to compute the

velocity vector at the center of pressure of each of these segments. This is accomplished by using velocities at each blade hinge point and trigonometric functions involving the blade lag and flap angles. Clearly, some of these equations must be calculated separately for each segment.

It is important to note that this simulation assumes rigid hinged blades which results in a great deal of simplification in the calculation of these velocity components.

#### Block 9: Rotational Equations of Motion for the Blade

The rotational equations of motion for each blade are obtained by summing moments acting on the blade about the hinge point. Here again, the blades are assumed rigid and only hinged connections are permitted in order to simplify the model. The outputs of this block are the blade span axes, roll and yaw rates which are obtained by integrating the corresponding accelerations. Separate equations are needed for each blade.

#### Block 10: Aerodynamic Forces on Blade Segments

This block is one of the major computational bottlenecks of the entire simulation. The output of this block are the forces along the blade axes. Their calculation requires evaluation of the respective lift and drag coefficients at each blade segment which in turn require the evaluation of the total velocity of each blade segment with respect to the inertial reference frame and the local angle of attack. Simplified versions of these equations were given in Chapter 2. The angle of attack and the Mach number are used as entry points to two-variable tables from which the lift and drag coefficients for each

segment are evaluated. The calculations for angle of attack, Mach number, lift and drag coefficients are performed separately for each segment of each blade.

In addition, this block includes a calculation of a geometric pitch angle for each blade segment which depends on this swash plate rotation (a control input) and various blade orientation angles. Uniform rotor downwash is calculated by applying momentum theory to the rotor thrust and then passing the results through a first order lag to approximate an air mass degree of freedom.

This rotor model differs from the C-81 model, largely from omission of the aeroelastic degrees of freedom, and by the assumption that the rotor blades are hinged.

#### Block 11: Moment Equations

The external moment vector consists of aerodynamic and hinge moments. The hinge moments are due to the spring damper constraints for the lagging and flapping degrees of freedom of each blade. The opposing moment due to the spring restraint is included as an arbitrary function.

#### Block 12: Equations for Computing Lag and Flap Velocities

The state variables  $\delta$  and  $\beta$ , which represent the lagging and flapping degrees of freedom respectively, are calculated from equations in this block. Clearly, separate equations are needed for each blade.

#### Block 13: Computation of Hinge Shear Force

Each blade has acting on it the sum of the gravity force, the aerodynamic force and the hinge shear force. The sum of these forces must equal the blade mass under the acceleration of each blades center

of gravity. In this block the hinge shear force is calculated for each blade in three axes.

#### Block 14 and 15: Total Rotor Force and Moment

The hinge shear force coefficients for each blade are used in the computation of the total rotor force and moment, which in turn are used as inputs to the fuselage dynamics in block 1.

#### 3.2. Digital Operations Involved in the Benchmark Problem

The number of digital operations required to solve the equations in the benchmark problem are summarized in Table 3.1. Note that separate counts are given for each block. For convenience, the operations are divided into multiplications or divisions, additions, trigonometric function generations, function generation, and square root operations. It is evident that the computational bottleneck lies in those blocks which are involved with computations for each segment of each blade, where functions of two variables must be evaluated. Table 3.1 is an overall summary of the operations required in each time frame for derivative evaluation. If a single pass integration formula is used, the bottom line in this table represents the actual number of computations per time frame. A higher order formula would require an appropriate multiple of the operations in this table.

Table 3.1 was used in obtaining the frame time estimates for various candidate computer systems which are discussed in Chapters 4 and 5.

It should be emphasized that the operation count described here, which represents a FORTRAN program of some 450 statements, does not include the control system, the fuselage aerodynamics or the engine.

However, the additional computation requirements are relatively small compared to the computational complexities involved in the evaluation of the rotor model.

### 3.3 Simulation Frame Time Requirements

In Section 3.2 the operation count associated with each major block of equations has been summarized, thus providing a guideline for the computing load. In this section we indicate the constraints concerning the time interval in which these computations must be performed.

Houck, in a careful study of the computational aspects of real time helicopter simulation, has investigated a variety of degradations of the model presented above to study their effect on the overall simulation process. In non-real time simulations, the rotor is represented by the actual number of blades, a fairly large number of blade segments sufficient to represent the blade loading accurately, the actual rotor rotational rate, and a sufficiently small azimuth advance angle per computational frame time. Note that the frame time for integration interval size can be expressed equivalently in terms of the azimuth advance angle per step. The simplifications which have been employed to achieve real time simulation have included various combinations of:

- a) reduced number of blades
- b) reduced number of blade segments
- c) reduction of rotational rate and
- d) large azimuth advance angle.

Typically, simulations at Sikorsky, Langley and Ames have used reduced models involving three rotor blades with three segments each and azimuth advances of the order of 50 degrees per frame. Even these degradations lead to computer frame times in the range of 40-50 milliseconds with

currently available computers at Ames Research Center. At the Langley Research Center using a Cyber 175, a 5-blade and 5-segment model with a 10 degree azimuth advance leads to frame times of the order of 10 milliseconds. This is probably the best available simulation with this order of complexity at the present time.

The maximum frame time requirement is easily calculated given a blade rotation speed and a desire for a 10 degree azimuth advance per frame. Assume that no aeroelastic degrees of freedom are included, in which case the highest frequencies would be flapping frequencies of one per revolution. If the rotor turns at 300 revolutions per minute this corresponds to a frequency of:

$$f = 300 \frac{\text{rev}}{\text{min}} \times 2 \pi \frac{\text{rad}}{\text{rev}} \times \frac{1}{60} \frac{\text{min}}{\text{sec}} \cong 31.4 \frac{\text{rad}}{\text{sec}} \quad (3.1)$$

If one now imposes the requirement of 10 degree per frame, the frame time  $T_f$  can be calculated as follows:

$$T_f = \frac{1}{31.4} \frac{\text{sec}}{\text{rad}} \times \frac{1}{57.3} \frac{\text{rad}}{\text{deg}} \times \frac{10 \text{ deg}}{\text{frame}} \cong .005 \text{ sec} \quad (3.2)$$

Hence, a 300 rpm rotor with a 10 degree advance requirement gives rise to a frame time of approximately 5 milliseconds. In the evaluations of candidate systems, 5 blades with 5 segments per blade were taken as the minimum simulation requirement, and 5 milliseconds was taken as the desired upper limit.

#### 3.4 Limitations and Extensions of the Benchmark Problem

It is evident that the current benchmark problem does not represent the potential upper limit of complexity which may be desired in rotorcraft

simulation. Helicopters with more complex rotor systems may involve more than 5 blades. Forward speeds of the order of 350 knots or even more are being suggested. Furthermore and most critically, the inclusion of aeroelastic degrees of freedom, both longitudinal and torsional, will greatly increase the computational complexity. If a blade bending degree of freedom with a frequency of 3 per revolution is included, the real time frequencies would now be of the order of 100 radians per second so that 5 millisecond frame times would correspond to approximately 30 degrees of motion in this bending mode or a sampling rate of approximately 12 samples per cycle. While this sampling rate is probably acceptable for minimum accuracy, it should be noted that the increased complexity of simulation arising from the addition of elastic degrees of freedom may give rise to as much as 25-50% more computation per frame. Hence, the authors accept as their goal the attainment of frame times in the range of 3 milliseconds for the benchmark problem.

No mention has been made in the above of attempts to include wake aerodynamics which are currently imperfectly understood and which could add additional computation complexity. However, it is anticipated that the additional computational load would be small compared to the addition of aeroelastic degrees of freedom.

### 3.5 Summary

In this Chapter the computational requirements associated with a specific (benchmark) helicopter simulation have been reviewed. It has been demonstrated that a minimal complexity rigid rotor simulation involving 5 blades with 5 segments per blade, requires frame times of the order of 5 milliseconds if the azimuth advance per step is to be restricted to 10 degrees.

TABLE 3.1

DIGITAL OPERATIONS COUNT FOR BENCHMARK ROTOR MODEL

Block#	Mult/Div	Add	Trig. Functions	One-Var. Functions	Two-Var. Functions	Square Roots
1	59	38	6			
2	8 + 11N	6 + 5N	6N			
3	8 + 2N	5 + 2N				
4	14	11				
5	20	19				
6	6N	4N				
7	11N	8N				
8	10N + 2Ns	5N + 2Ns				
9	14N	9N				
10	6N + 26Ns	5N + 6Ns	2N + 2Ns		3Ns	3Ns
11	4N + 2Ns	2N + 2Ns		N		
12	N	2N				
13	10N	6N + 3Ns				
14	8 + 12N	5 + 10N				
15	14 + 5N	11 + 5N				
TOTAL	131+92N+30Ns	95+63N+ 13Ns	6+8N+2Ns	N	3Ns	3Ns

Total for

N=5,s=5    1341                    735                    96                    5                    75                    75

N = Number of Blades

s = Number of Blade Segments

## IV ALTERNATIVE ARCHITECTURAL APPROACHES

### 4.1 General Purpose vs. Special Purpose Processors

To date all digital simulations of complex aerospace systems, including the helicopter problem have been carried out on large general-purpose digital computers. All of the differential equations characterizing aircraft dynamics and control, have been programmed on such computers, which also carry out all communication and system control functions. General-purpose computers which have been used in this way at various locations include the Control Data 7600 and CYBER 175, as well as the larger models of the IBM 360 and 370 series. These machines have proved to be adequate for most simulations, but as the complexity of proposed simulations increased their performance has become more and more marginal. For example, the CYBER 175 at NASA/Langley can handle helicopter simulations only in a simplified form. A new generation of large general-purpose computers, sometimes termed super-computers, has recently been introduced. Machines of this type, including the STAR and the CRAY-1, show promise of providing speed sufficient for even the most ambitious simulations, however at a very high cost.

An alternative approach to the attainment of very high-speed real-time capability is to fashion a computer system by interconnecting a moderately sized general-purpose digital computer and a special-purpose digital processor, as shown in Figure 4.1. The host computer which acts primarily as a buffer to the communication lines and input/output equipment can be a large minicomputer such as the Digital Equipment Computer PDP-11/70. The peripheral processor is a digital computer

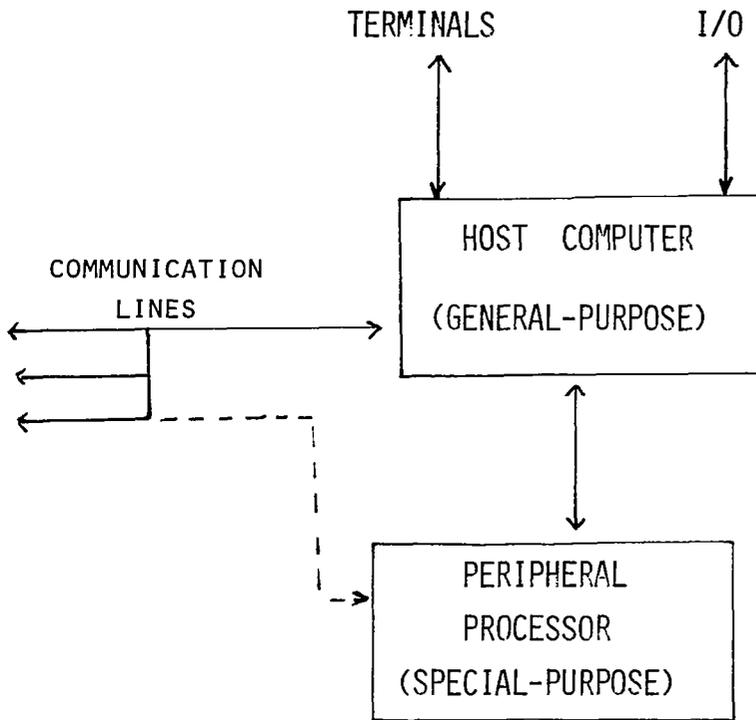


FIG. 4.1

employing a high degree of pipelining and parallelism. Because it is designed solely to facilitate high-speed arithmetic computation, it is capable of providing computing speeds considerably superior to those of large general-purpose computers at a very moderate cost. Thus while large general-purpose computers cost well over \$2,000,000 and super-computers over \$8,000,000, a system of the type shown in Figure 4.1 can be acquired for less than \$300,000. To be sure, host/peripheral systems are much more difficult to program, particularly if optimum or near optimum operation is desired. Remarkable progress has been made during the past two years in the hardware design of peripheral processors and in the development of software packages to facilitate their programming. Some of the more promising of these configurations are considered in Sections 4.2, 4.3 and 4.4.

#### 4.2 Array Processors

Array processors constitute a family of peripheral processors which has been developed in recent years for signal processing applications. There exists a great need, particularly in the petroleum and the medical fields, for high-speed frequency analysis, often employing Fast Fourier Transformations. Peripheral processors for this application are generally structured as shown in Figure 4.2. They provide a bus-structure which permits the simultaneous performance of fetches from a data memory, multiplication and addition. Data to be processed, for example several thousand samples of a continuous seismic or biological signal, are read into the peripheral processor from a host computer and subjected to a very rapid sequence of arithmetic operations. Through the use of the latest solid-state circuitry, compact design, and extensive pipelining, these

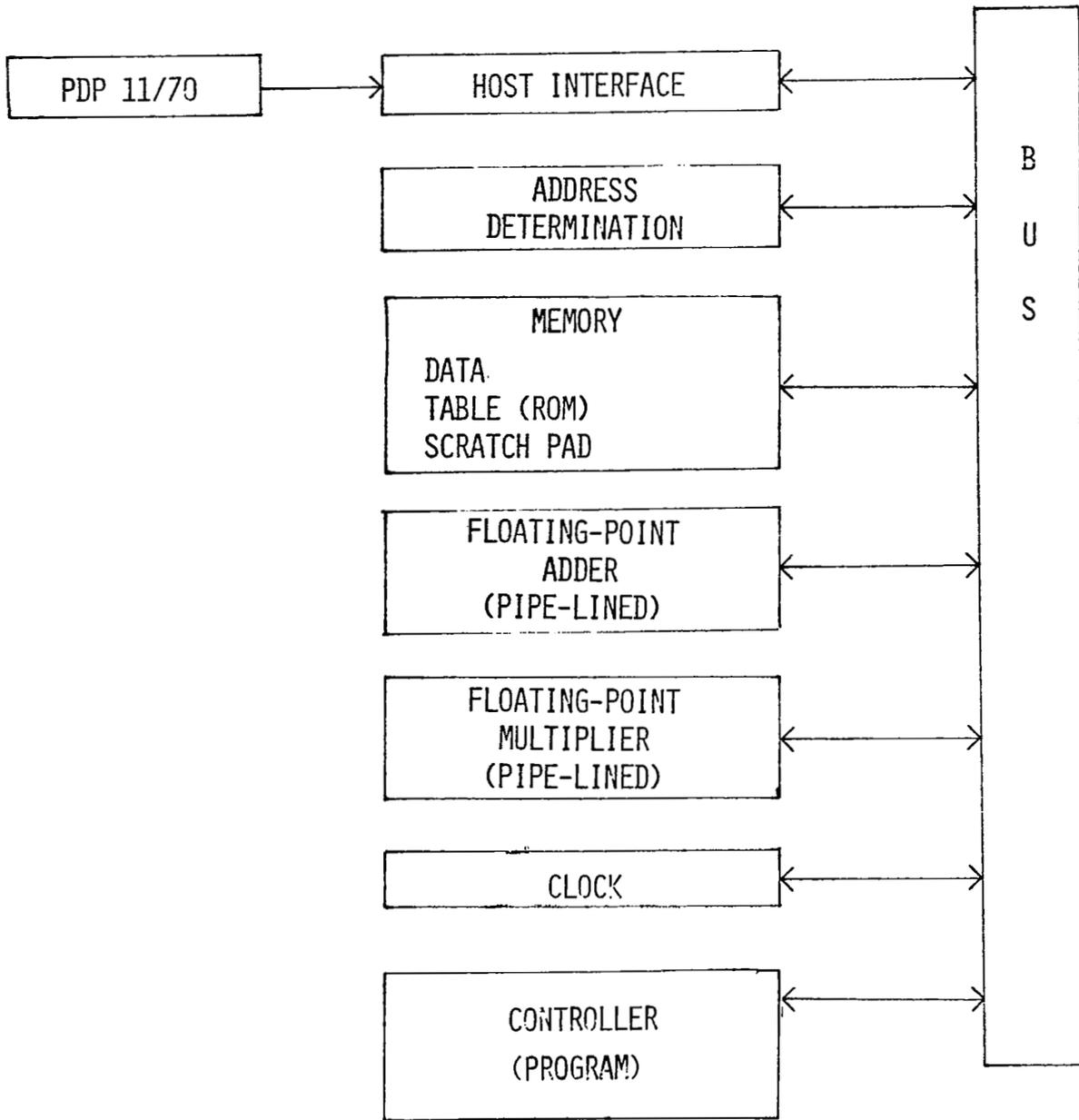


FIG. 4.2 ARRAY PROCESSOR

peripheral processors are able to exceed the speed of even the most powerful general-purpose computers at a relatively modest cost.

Among the array processors which have been introduced during the past two years are two systems intended to enhance performance of large computers. The Control Data Corporation MAP-3 is designed to serve as an adjunct for the CYBER computers. Similarly the IBM 3838 is designed to complement the IBM 370 series. Other array processors are intended to be used in conjunction with minicomputers. Examples of this type of peripheral include the SPS-41 of Signal Processing Inc., the Real-Time I, II and III Systems of Datawest Corp., the MAP series of CSP, Inc., and the AP-120B of Floating Point Systems, Inc. Of these, the last mentioned unit has seen the most wide use.

#### 4.3 Simulation-Oriented Peripheral Processors

The special-purpose digital processors of this class have an architecture very similar to that of the array processors. However, parallel units are included to facilitate those tasks which are peculiar to simulation, notably function generation and integration. Figure 4.3 is a general block diagram of such a system manufactured by Applied Dynamics Inc. It contains separate units for breakpoint determination, interpolation, and memory mapping - all required as part of the generation of functions of two or more variables. Such a system is capable of all of the arithmetic operations of array processors and provides additional capabilities useful in real-time aircraft simulations.

#### 4.4 Arrays of Processors

The prospect of fashioning networks of general-purpose computers so as to obtain greater speed through parallelism has been a tantalizing

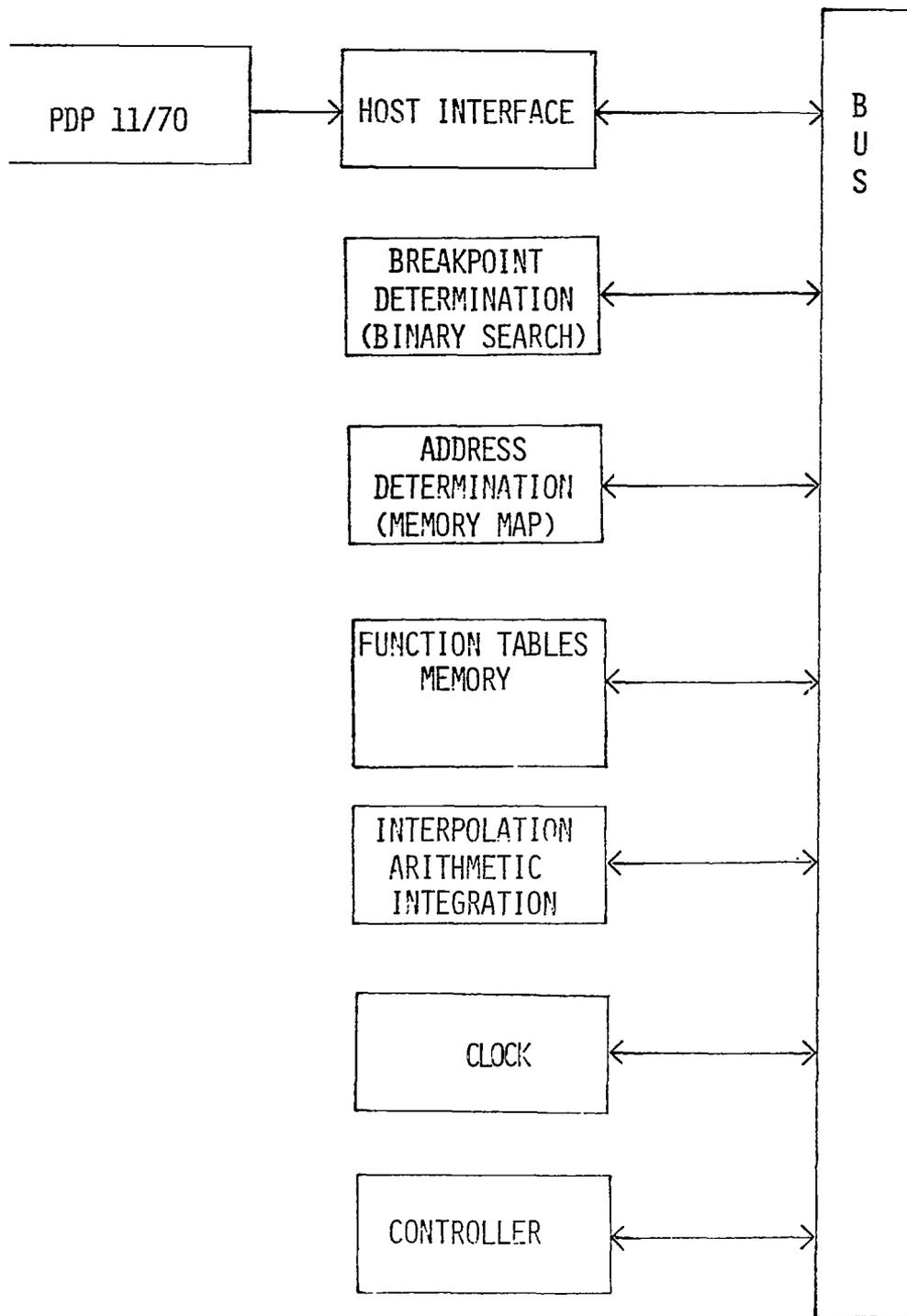


FIG. 4.3

prospect for many years. ILLIAC IV has the general-structure shown in Figure 4.4. A single control unit supervises the operation of a large number of parallel-operating arithmetic and logic units, each with its own memory. The optimum or near optimum operation of such a system requires highly advanced hardware and software techniques. During the past year, the advent of reliable ECL integrated circuits and new design techniques make it possible to contemplate arrays of processors of very high speed and at a cost and reliability far superior to that of ILLIAC IV. The G-471 proposed by W. W. Gaertner Inc. and the HEP now under construction by Denelcor Inc. are modern examples of this approach.

Professor G. A. Korn has for some time championed the fashioning of rapid and inexpensive digital simulators by interconnecting general-purpose minicomputers. He has pointed out that a system consisting of three Digital Equipment Corp. PDP-11/45's can outperform analog computers in most applications. To date, no large scale simulations of this type have been undertaken, and some knotty software problems remain to be resolved.

#### 4.5 Hybrid Computers

For many years, analog and hybrid computers were the only vehicle for real-time simulation. By performing integrations using analog integrators, truncation errors and round-off errors are completely obviated. Because analog devices are inefficient when it comes to logic operations and the generations of complex nonlinear functions, analog systems are usually connected in a closed loop with a digital computer, thereby forming hybrid computer systems. The two leading American manufacturers of general-purpose hybrid computers are Electronic Associates, Inc. and Applied Dynamics, Inc. Both of these manufacturers have an

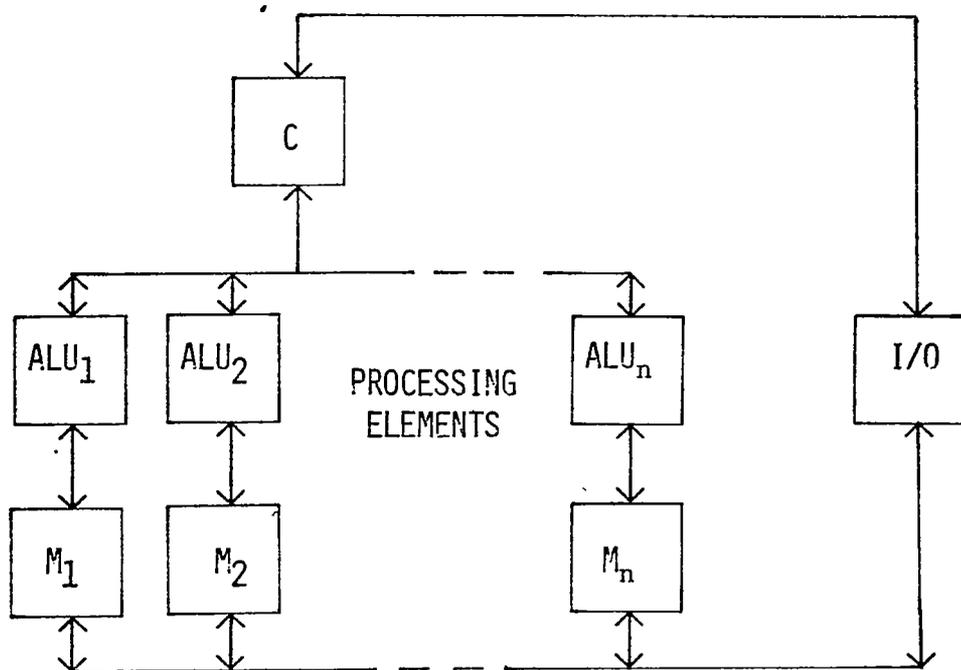


FIG. 4.4

impressive product line of modern hybrid computers. A major difficulty with these systems is that they are relatively difficult and awkward to program and to set up. Although automatic patchboards have been introduced for small problems, they have not been refined to the point that all large problems can be programmed automatically. Furthermore a completely expanded hybrid computer system costs well over \$500,000 and requires extensive and specialized maintenance. For these and other reasons, more and more simulation facilities have been turning to digital simulators in recent years

Commercially available hybrid computer systems are expensive because they are designed to be relatively general-purpose in nature. Great savings in cost and space can be effected by designing a hardwired analog or hybrid system for a specific application. This is the approach taken by Paragon Pacific, Inc. in the design of the SPURS system for helicopter simulations. This system is available at a fraction of the cost of general-purpose hybrid computers and has a superior dynamic performance. It can only be used for a specific problem, however; and major changes in the model require an extensive rewiring of the unit.

#### 4.6 Other Possibilities

Digital differential analyzers are a special type of digital computer especially designed for the solution of differential equations. Data is transferred within the machine not as whole numbers, as in general-purpose digital computers, but as single-bit increments. While relatively inexpensive and fast, digital differential analyzers have in the past suffered from a variety of error and reliability problems. Recently developed integrated circuitry may finally permit the fashioning

of truly competitive digital differential analyzers. However, no vendor in this country has as yet placed such systems on the market nor have advanced software packages been developed.

Probe Systems, Inc. has suggested the development of a special purpose computer system oriented toward simulation. Their concept is novel in that it is intended to facilitate the programming of the simulator in FORTRAN rather than in a lower-level language. This would be an appealing feature. No details of the approach to be used have been provided however.

## V. DESCRIPTION OF LEADING CANDIDATES

The helicopter real-time simulation employed as a benchmark in the present study, as well as other aerospace problems of comparable complexity, constitute a severe challenge to even the largest of present day general-purpose digital computers.

If the simulator consists of a general-purpose computer augmented by a peripheral processor, the problem becomes readily tractable. In fact, a number of the approaches described in Chapter IV, involving the combination of a large-mini-computer and a special-purpose peripheral unit, are potentially capable of achieving frame times of 0.5 to 1.0 millisecond. The overall hardware costs for such systems would be of the order of \$200,000 to \$300,000 including the cost of a general-purpose computer such as the PDP-11/70. To be sure, some knotty software problems have to be resolved, and the system would be less flexible and convenient to use than would a large or super-computer.

In this chapter five relatively promising hardware systems are examined in some detail. The hardware and software attributes of each system are considered in turn together with a calculation of its speed in solving the benchmark problem. For purposes of comparison, the computing times required by each candidate to carry out the digital operations listed in Table 3.1 have been calculated. It must be emphasized that the overall frame time achievable in actual simulations is actually the time required for data communication and housekeeping in the host computer plus the time required to solve the differential equations. This total time is probably well in excess of 10 milliseconds and may therefore completely overshadow the frame times presented below.

## 5.1 Floating Point Systems, Inc. - AP-120B

### Overview

The AP-120B belongs to the family of array processors. During the last two years at least six new models of such devices have been introduced. All were designed primarily to compete for the growing market for signal processors, particularly in the seismology and medical image processing fields; and all are roughly similar in architecture and performance. The Floating Point System unit was selected for detailed consideration because it is the most widely used device (over 150 have been delivered), and because an impressive variety of software packages are available. Moreover it is readily capable of being interfaced with minicomputers such as the PDP-11/70 - this in distinction to the IBM 3838 and the CDC MAP III which are designed primarily to interface large IBM and CDC computers.

Though not as fast as some of the other candidates, the AP-120B in conjunction with a PDP-11/70 can provide a frame time of 2.0 to 3.5 milliseconds for the benchmark problem, depending upon the algorithm employed for function generation. The cost of the unit with the required memory expansion would be about \$85,000. Currently available software packages include a convenient assembler, extensive subroutine libraries, and a simulator. The latter permits the AP-120B to be emulated on a general-purpose computer, of course at a greatly reduced speed.

### Hardware Organization

The AP-120B is mounted on a standard 19" rack and requires approximately two feet of vertical space. The power supply is located directly behind the forward unit which contains all other circuitry. There is

provision in this forward unit of up to 28 15" x 10" circuit boards which plug into a mother board. These circuit boards are filled with standard LSI units. Figure 5.1 is a block diagram of the forward unit.

The system elements are interconnected with multiple paths so that transfers can occur in parallel. All internal floating point data paths are 38 bits in width (10-bit exponent and 28-bit mantissa). The interface unit is designed especially for the host computer and is organized so that either I/O or the DMA channels can be utilized for data transfer. Instruction and data transfers take place at a 6 MHz rate, corresponding to a cycle time of 167 nanoseconds.

The operation of the unit is controlled by the execution of 64-bit instruction words which reside in the program memory. Access to the program memory and instruction decoding are overlapped so that the unit can operate at the 6 MHz clock rate. Additional control functions are provided by the S-PAD unit which performs integer address indexing, loop counting and other control functions required by specific algorithms. The S-PAD contains sixteen 16-bit directly addressable registers whose contents pass through a special integer arithmetic and logic unit.

The floating point adder does addition or subtraction operations on the contents of the adder input registers A1 and A2. The operation is performed in two stages each of which takes one machine cycle or 167 ns. Since the two stages are independent of each other, a new pair of numbers may be entered into the input registers every machine cycle providing for pipeline operation.

The floating point multiplier generates the product of the contents of the two input registers M1 and M2. This product is formed in three

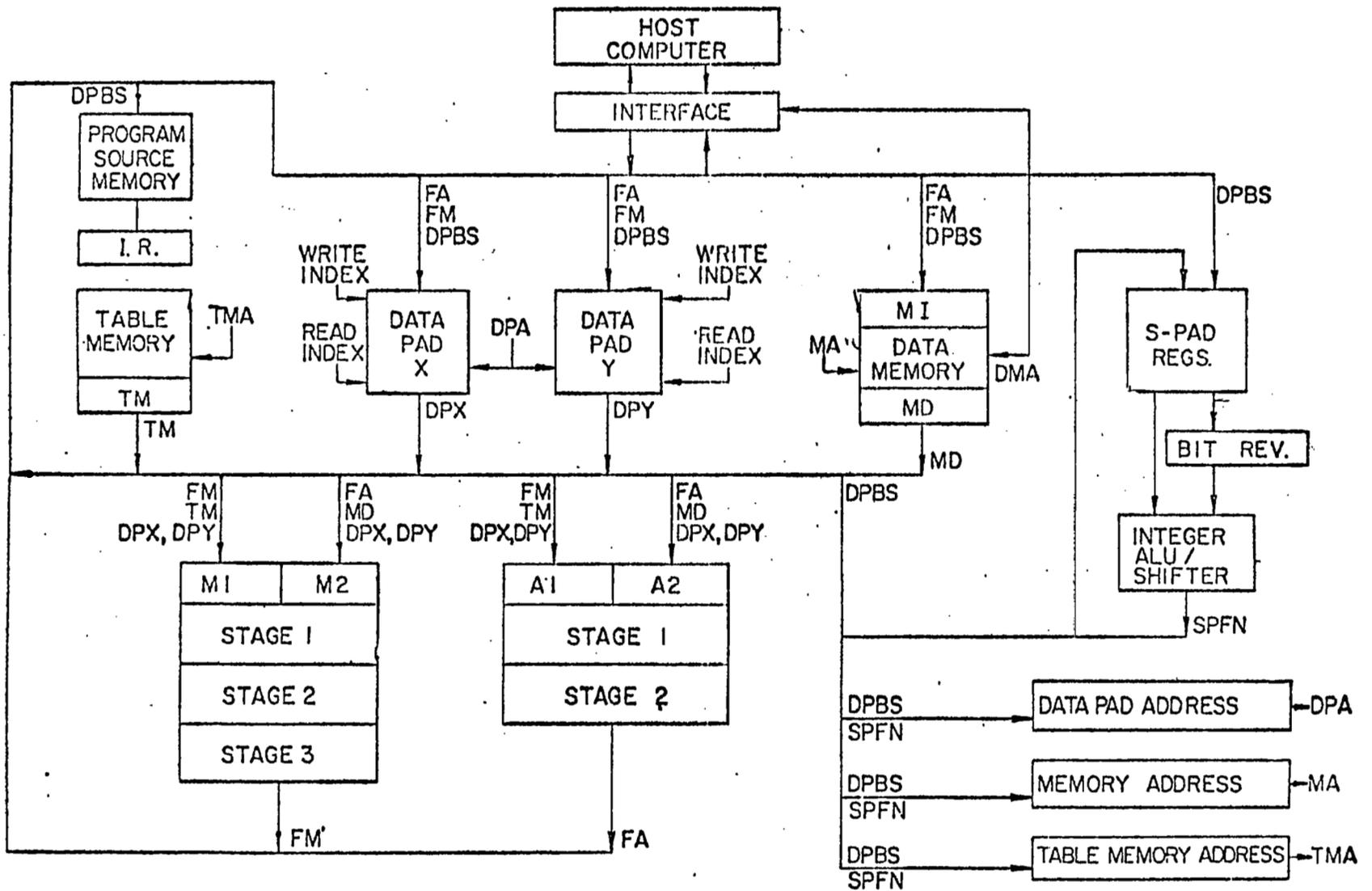


Fig. 5.1 AP-120B Block Diagram

stages, each of which requires 167 ns. A new product may thus be started every 167 ns, but the result is not obtained until 500 ns later.

The data pad unit consists of two fast accumulator blocks, each with 32 floating point locations. In a single machine cycle, the contents of one location from each of the two blocks may be read out and used. In addition, data may also be read into one location in each block in the same cycle. This unit serves primarily for the storage of intermediate results of computations.

The data memory unit is the primary data store for the AP-120B. It is available in 38-bit wide 8K modules which have an interleaved time of 333 ns. For reasons of economy this unit is fashioned from MOS integrated circuitry, while bipolar circuitry is used elsewhere in the processor. A memory operation may be initiated every other machine cycle. To optimize the operation of the processor, it is necessary for the programmer to "look ahead" and initiate memory reads prior to the actual time that arguments from data memory are to be used in the calculation.

The table memory unit employs rapid and therefore more expensive circuitry and, as the name implies, it is used to store data for table look-up utilization. A new table value may be requested every machine cycle.

#### Mode of Operation

Prior to the computer run, the program for the entire computation to be performed in the AP-120B must be loaded into the program memory. Presumably this program is resident in one of the auxiliary memory units of the host computer. The maximum size of the program memory is 4000 words, which places a ceiling on the computations performed during

each time frame. On the other hand, each program word actually contains six parallel 10-bit instructions to the different parallel-operating units, so that this memory actually houses considerably more than the equivalent of 4000 assembly language instructions. Also prior to the commencement of the run, the table memory has been loaded with all required tables. These include the trigonometric functions, the aerodynamic functions (lift and drag coefficients versus Mach number or angle of attack) as well as other tables designed to minimize the computation. This may include tables for  $x(x^2 + y^2)^{-1/2}$ , reciprocals, etc.

If it is assumed that the control system is represented in the host computer, while all differential equations governing rotor dynamics are solved in the AP-120B, the inputs to the peripheral processor at the beginning of each time frame include rotor control signals, the deflection angles for the stabilizing surfaces and changes in the magnitude of the thrust vector. Environmental effects, failure modes and other changes to the simulation will also be transmitted from the host computer. These terms may be read out of the host computer via the DMA channel and placed in the data memory. Because the input terms are so few in number however, it may be more efficient to transfer them at once to the data pad or random access table memories. Under these conditions the data memory would not be used at all.

The program resident in the program memory is then executed without further communication with the host computer. At the end of each time frame all the quantities needed for cockpit instrument displays and for control system computations are transmitted to the host computer. These variables are then read into the host computer via the interface unit and the DMA channel.

In essence, therefore, from the point of view of the host computer the AP-120B acts as a subroutine which is called once during each time frame.

### Speed

As is the case for most peripheral processors, it is advantageous to minimize transfers across the interface. Accordingly, it has been assumed that only the control calculations will be performed in the host computer, while all other calculations are performed in the AP-120B. It is further assumed, that the integration algorithm to be used requires but a single function evaluation during each time frame.

The computer times required for the various steps involved in the solution of a single time frame of the benchmark problem are shown in Table 5.1. The following basic assumptions have been made in preparing this table.

1. Each multiplication requires three clock pulses or 0.500 microsecond.
2. Each addition requires two clock pulses or 0.333 microsecond.
3. Functions of a single variable (particularly sines and cosines) require 42 clock pulses for binary search and 10 clock pulses for interpolation for a total of 8.63 microseconds.
4. Functions of two variables require 54 clock pulses for binary search and 20 clock pulses for interpolation or a total of 11.84 microseconds. This assumes that each function of two variables is presented as a 32 x 16 array.
5. Terms such as  $x(x^2 + y^2)^{-1/2}$  are tabulated as two-dimensional arrays.

If N represents the number of rotor blades, and s is the number of finite elements per blade, the total time, T, required for a time frame

TABLE 5.1

FRAME TIME OF THE FPS AP-120B FOR  
THE BENCHMARK PROBLEM

	<u>Time Required</u> <u>μsec</u>	
Computations Independent of N and s		
131 Multiplications	65.5	
95 Additions	31.6	
6 Functions of One Variable	<u>51.7</u>	148.8 μsec
Computations Proportional to N		
92 Multiplications	46.0N	
63 Additions	21.0N	
9 Functions of One Variable	<u>77.6N</u>	144.6N μsec
Computations Proportional to N and s		
30 Multiplications	15Ns	
13 Additions	4.3Ns	
2 Functions of One Variable	17.2Ns	
6 Functions of Two Variables	<u>71.0Ns</u>	107.5Ns μsec
Time for N x s = 25		3.56 milliseconds
N = number of rotor blades	s - finite elements per blade	

is

$$T = 148.8 + 144.6 N + 107.5 Ns \quad (5.1)$$

where the times are expressed in microseconds. For a helicopter with five rotor blades and five segments per blade, the frame time becomes 3.56 milliseconds.

This calculation is conservative for several reasons. In determining the total time, no attempt has been made to take advantage of the parallelism or pipelining available in the AP-120B. With reasonably clever programming a substantial saving could be effected. Also, all function evaluations are assumed to involve a substantial binary search. In fact for the 25 element rotor, 2.03 milliseconds are devoted to the binary search. Such a search is actually required only if the elements of the array being searched are nonuniformly spaced. If the table is made large enough to permit the uniform spacing of all elements, a direct address calculation can be performed in lieu of the binary search. In this way, the frame time can be reduced by at least 1.0 millisecond.

#### Programmability

Of all of the leading candidates, the AP-120B has by far the most impressive software support. All of the programming packages have had extensive use and are capable of running on any computer with a FORTRAN compiler. The available software packages fall into four major categories.

Executive routines

Mathematical Library of subroutine calls

Program Development Packages

Debug programs

The executive program (APEX) is a mechanism for communicating with the AP-120B via a series of FORTRAN or machine language subroutine calls. The AP-120B is capable of operating under the standard operating systems of most host computers including the UNEX program of the PDP-11 series.

The mathematics library includes 70 subroutines written in AP-120B assembly language. These are callable from the host computer as required. Most of the available subroutines are intended primarily for signal processing applications such as Fast Fourier Transforms and would probably not be used in aerospace applications. All of the transcendental functions required in aerospace work are available.

The Program Development Package includes an assembler, a linker, a debugger, and a simulator. All of these are written in FORTRAN and compiled on the host computer. The assembler provides a two-pass assembly of symbolic coding into an object module, and also generates detailed error diagnostics. Using this assembler, programming the AP-120B entails the preparation of a separate instruction word for each machine cycle. Each word contains separate instructions to each of the six subunits. Some effort is of course required to become familiar with the assembly language and to optimize program structure.

The simulator portion of the Program Development Package provides a program simulation of the various hardware elements of the AP-120B. All timing characteristics of the AP-120B are emulated, and the floating point arithmetic is simulated (including rounding) to the least significant bit. Using this package, new AP-120B programs can be developed off-line, even at distant and independent computer facilities.

There is no question that the utilization of the AP-120B entails

programming skills and special knowledge far beyond that required in the programming of similar problems on large general purpose machines. It is unlikely that there will ever be available sufficiently powerful compilers to permit the host computer/AP-120B system to be programmed in FORTRAN, without sacrificing most of the speed advantages. On the other hand, the availability of the assembler and the simulator greatly facilitates the programming task. No doubt, the system will rarely be operated at maximum efficiency; but it does not appear that an extraordinary programming effort would be required to prepare application programs with adequate speed.

#### Application Support

The AP-120B was developed primarily with signal processing in mind. Floating Point Systems, Inc. maintains a competent staff of programmers to provide programming support. However, simulation is far removed from the bread-and-butter activities of this group, and it is unlikely that there is sufficient incentive for FPS to develop any appreciable capability in that direction. It follows therefore that FPS will give negligible support as far as specific simulation applications are concerned.

#### Principal Shortcomings

The AP-120B is a machine that has clearly been developed for an application other than simulation. The fact that it appears to be adequate for the benchmark problem may be considered a happy accident rather than the result of planning. It cannot be assumed therefore that other aerospace problems can be accommodated as readily as the helicopter benchmark problem.

The AP-120B was designed primarily for problems in which large blocks of data are read from the host computer into the data memory, and in which the elements of these blocks are then subjected to a relatively short series of arithmetic operations and manipulations. This is the situation in signal analysis and Fast Fourier Transformation. In aerospace simulation problems on the other hand, a relatively small sequence of numbers (typically twelve or less) are read out of the host computer at the beginning of each time frame. These data are then subjected to extremely lengthy and elaborate computations. For example in the helicopter problem, the input vector only contains ten elements. The FORTRAN program describing the manipulations of this vector during each time frame require over 450 FORTRAN words, not counting comments, declarations, etc. As a matter of fact, in the solution of the helicopter problem on the AP-120B it is possible that the data memory will not be used at all; in signal processing applications, on the other hand, the data memory is the central element for all operations. The program memory can be expanded to 4000 64-bit words. It is a complicated matter to predict how the 450 FORTRAN commands would translate into AP-120B assembly and machine language. For the benchmark problem, the program memory would probably be sufficiently large, but it is quite possible that other aerospace simulations may have excessive program memory requirements. The implications involved deserve considerably more detailed study.

Another disadvantage is the absence of facilities for direct access to the AP-120B from external communication lines. All data must enter and leave the unit via the interface module and the host computer. This

may create an intolerable bottleneck under certain conditions.

#### Credibility of the Unit

By the end of 1976, over 150 AP-120B units had been delivered. The majority of these were destined for OEM applications, particularly in the seismic and medical fields. As would be expected of an item which accounts for the bulk of a company's sales, a large effort has gone into optimizing all its hardware and software features. Only standard high-quality modules, obtained from major suppliers such as Texas Instruments Inc. are employed, and impressive quality control techniques are in regular use. The mean time between hardware failure of this unit was reported to be 3800 hours - a very noteworthy record. If the AP-120B is acquired it can be assumed that it will be more reliable and cause fewer headaches than the host computer and its other peripherals.

#### Costs

With a fully expanded Program Source Memory and a reasonably extensive Table Memory, the AP-120B for the present application would come to approximately \$80,000. This figure includes the necessary software packages and an interface to the host computer.

## 5.2 Applied Dynamics Inc. AD-10

### Overview

The AD-10 was originally designed to serve as a digital function generator for hybrid computer systems. Only subsequently were the potentialities of this unit as a peripheral for general-purpose digital computers recognized. The architecture and mode of use of this unit in the present application is essentially similar to that of the family of array processors. The subunits making up the AD-10 are however in many ways more suitable for simulation applications. In fact the simulation orientation of the AD-10 is one of its more attractive features.

In order to make the AD-10 suitable for the present application, it will be necessary to augment it with an integrator module, which has been designed on paper but not yet built. With this module, the AD-10 in conjunction with a PDP-11/70 can provide a frame time of approximately 0.6 millisecond for the benchmark problem. The cost of the unit with the required features would be about \$85,000. An assembler is currently available, but a more sophisticated FORTRAN oriented version will have to be provided by the vendor. The AD-10 is a relatively new machine. Several prototypes exist, but none have actually been used in practice.

The use of the AD-10 as a peripheral to a digital computer entails two major difficulties. Except for the integrator module which has a 48-bit word, the AD-10 is a 16-bit machine. This should be adequate for most applications but may occasionally present difficulties. More seriously, the AD-10 operates in the fixed-point mode. This implies

that all computations within the AD-10 must be carefully scaled to avoid overflow or loss of significant figures. A careful and detailed study will be required to determine to what extent this negative feature is overshadowed by the positive features of the AD-10.

#### Hardware Organization

The AD-10 occupies most of a standard rack. Because ECL circuitry is used extensively, power requirements are substantial and a large power supply is provided. Figure 5.2 is a block diagram of the AD-10. The data, address, and control multibus is a parallel ECL bus, composed of 16 data lines, 18 address lines and several control lines. This bus supports twenty data/address bus transactions per microsecond. These bus transactions pass data to and from the data memory, the functional processors, and the host digital computers. The transactions as well as all memory processor functions are synchronously controlled by a master 40 MHz clock.

The AD-10 is unique among the peripheral processors encountered in this survey in that it has a distributed program control memory. Each functional unit has a separate instruction memory controlling the actions of the corresponding functional unit each machine cycle. Prior to a computer run, each of these program memories is loaded by the host computer. The host computer is coupled to the AD-10 via the Host Interface Controller which distributes the instructions to the appropriate functional processors and loads function data into the Multiport Data Memory.

The Multiport Data Memory holds all tabular data as well as break-point values and slope/gain factors for multivariable functions. All

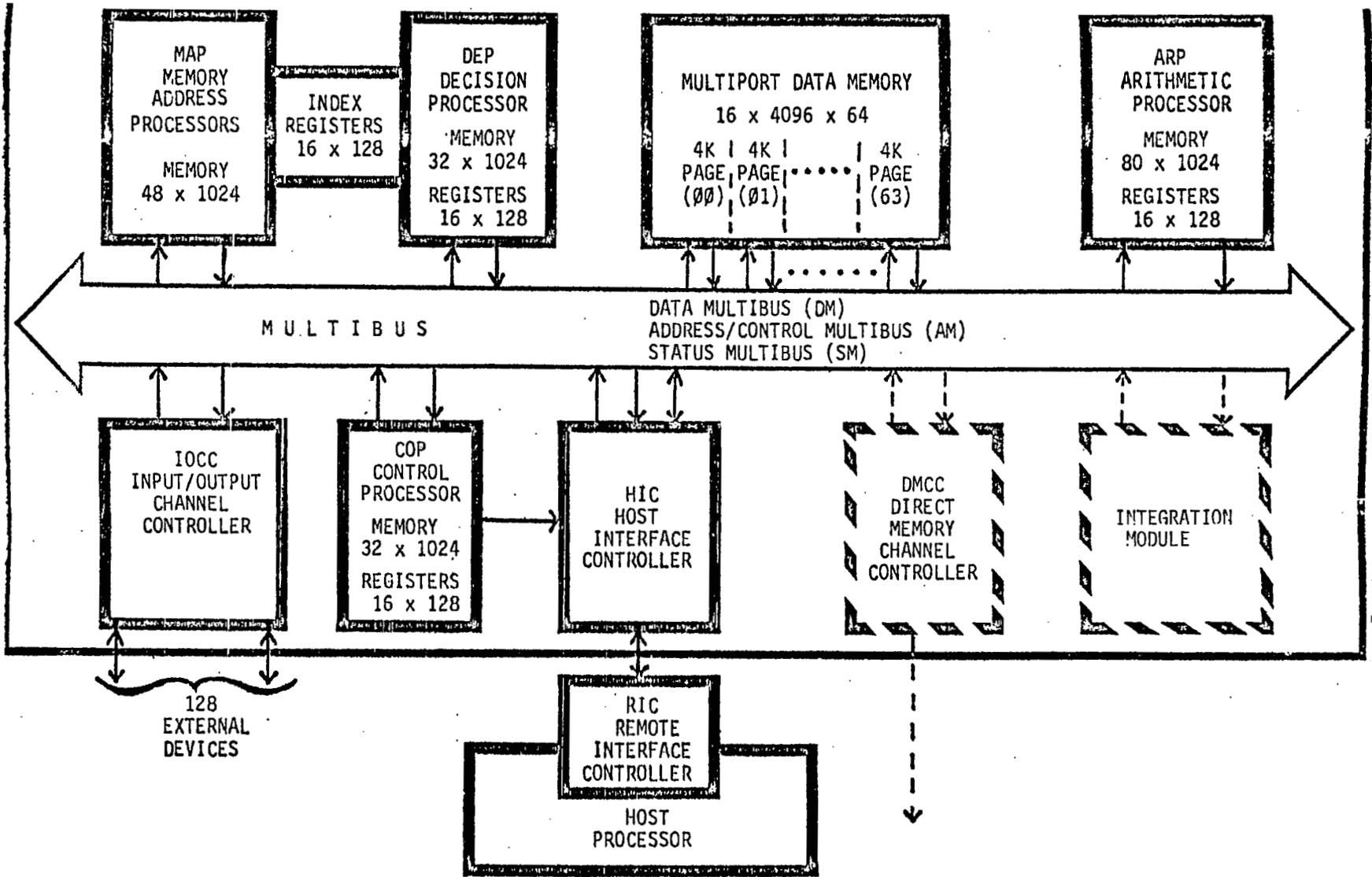


Fig. 5.2 Applied Dynamics AD-10

function data are organized in the data memory so that data fetches may be accomplished at a 20 MHz rate. The memory is organized in pages of 4096 words, and each page is ported separately so that it may be addressed independently.

The Memory Address Processor generates physical addresses for the Data Memory from virtual addresses and therefore acts essentially as a memory map. The Decision Processor efficiently implements a binary search for breakpoint values as well as other decision-oriented operations. By isolating these decision operations within this unit and the Memory Address Processor, the Arithmetic Processor is simplified and addressing is made more efficient. This is a particularly important feature in function generation.

The Arithmetic Processor provides high-speed arithmetic capability. The speed of this unit results from the use of pipelining techniques, overlapped move and arithmetic operations, and the inclusion of a very-fast 128-word temporary register file. Figure 5.3 is a block diagram of the arithmetic processor unit. This unit is designed to execute an arithmetic instruction of a general form

$$R = \pm (A \pm B) * C \pm D$$

in 175 ns. Of course these are fixed-point 16-bit operations. The unit is seen to contain two adders and one multiplier as well as a number of temporary storage registers. Since all the operations are fully pipelined, 20 additions and 10 multiplications can be achieved in a single microsecond. Error messages are generated if there are out-of-range errors in the positive or negative direction. The arithmetic processor program memory contains 1024 words. Each 80-bit instruction

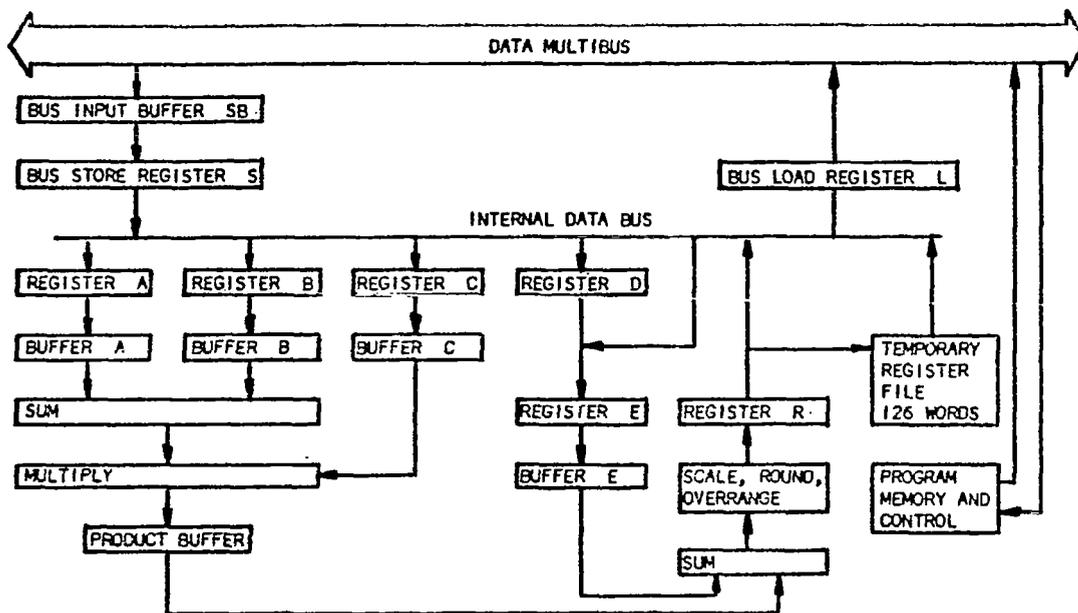


Fig. 5.3 AD-10 Arithmetic Processor

word is divided into five 16-bit fields, and each field is individually addressable.

In order to minimize data transfer across the interface controller during each time frame, it is essential that the AD-10 be equipped with a facility for integration. This capability was omitted from the original design of the AD-10, since it was intended that the AD-10 function in conjunction with analog integrators. Applied Dynamics has completed the preliminary design of an integrator module and expects that this unit will be operational in the latter part of 1977. Unlike the other functional modules of the AD-10, the integrator module employs 48-bit rather than 16-bit words. In that way, the deleterious effects of round-off error accumulation can be avoided. The design of this

TABLE 5.2

FRAME TIME OF THE APPLIED DYNAMICS AD-10 FOR THE BENCHMARK PROBLEM

	Time Required $\mu\text{sec}$	
Computations Independent of N and s		
131 Multiplications	33.9	
95 Additions		
6 Functions of One Variable	17.2	
		51.1 $\mu\text{sec}$
Computations Proportional to N		
92 Multiplications	23.3 N	
63 Additions		
9 Functions of One Variable	9.9 N	
		33.2 N $\mu\text{sec}$
Computations Proportional to N and s		
30 Multiplications	6.5 Ns	
13 Additions		
2 Functions of One Variable	2.2 Ns	
6 Functions of Two Variables	7.8 Ns	
		16.5 Ns $\mu\text{sec}$
Time for N s = 25		0.63 millisecc
N = number of rotor blades	s = finite elements per blade	

module appears reasonable, and it is unlikely that major difficulties will be encountered in its implementation.

### Speed

With the forthcoming availability of an integrator module, all differential equations governing the dynamics of the helicopter can be solved in the AD-10 during each time frame. As in the case of the AP-120B, it is assumed that the integration algorithm selected requires but a single function evaluation during each time frame. Under these conditions the computer times required for the various steps involved in the solution of one time frame of the benchmark problem are as listed in Table 5.2.

Because of the unique organization of the Arithmetic Processing unit some overlap of multiplication and addition operations have been assumed. Additionally, it has been assumed that a full binary search is required for function generation.

If  $N$  represents the number of rotor blades, and  $s$  is the number of finite elements per blade, the total time  $T$  in microseconds for a single time frame is

$$T = 51.1 + 33.2 N + 16.5 Ns \quad (5.2)$$

for a helicopter with five rotor blades and five segments per blade, the frame time becomes 0.63 millisecond.

This time is probably conservative, since approximately 0.15 millisecond can be saved if the need for a binary search is obviated by spacing the elements of the functional arrays uniformly. Also some additional time can be saved by fully overlapping multiplications and additions and keeping the pipeline full at all times.

### Mode of Operation

Prior to a computer run, the program memory of all of the functional units must be loaded. Presumably these programs are resident in one of the backup memory units of the host computer. These programs are transferred to the functional unit, via the Interface Controller. All tabular information is also loaded from the host computer via the Interface Controller into the Multiported Data Memory. These tabular data would include all trigonometric functions as well as a number of other combinations of variables. Since the AD-10 is particularly efficient in table lookup, it is expedient to avoid all computations of square roots, reciprocals, etc. by storing these functions in tabular form.

In applying the AD-10 to the helicopter problem, all computations involving the solution of the differential equations governing the system dynamics are solved in the AD-10. At the beginning of each time frame, control signals to the rotor, stabilizing surfaces and engine, as well as environmental changes, are transferred from the host computer to the AD-10 via a direct memory access channel. The program resident in the AD-10 is then executed independently of the host computer. At the end of the time frame, the variables needed for cockpit displays and control system computations are read out of the AD-10 and into the host computer. Aircraft control system functions and other operations requiring interaction with the outside world are implemented on the host computer. From the point of view of the host computer then, the AD-10 appears as

a differential equation solving subroutine which is called once during each time frame.

#### Software Support

Relatively meager software support exists for the AD-10 at the present time. A preliminary version of an assembler is in existence and appears to be operating satisfactorily. Applied Dynamics is committed to providing a FORTRAN based assembler more suitable for general use. A library of subroutines will also be available in the near future. With these software packages, the preparation of application programs for the AD-10 will not be extremely difficult. It will however be necessary to prepare the program one machine cycle at a time and to specify the actions of each of the functional units. Each of the functional units has the capability of pausing while other units catch up with their computation. It is not necessary, therefore, to strive for an optimum. Adequate frame times appear to be feasible using a relatively unsophisticated approach.

#### Application Support

Applied Dynamics has been in the simulation business for over 15 years. Members of its staff are among the leading experts in the mathematical modeling and simulation of aerospace systems. This experience has been brought to bear on the design of the AD-10. It will also be invaluable in guiding the development of useful software packages and in providing backup for customer application programming.

#### Principal Shortcomings

When augmented with the integrator module now under construction, the AD-10 will constitute an exceptionally powerful peripheral to a host computer. At the present time, insufficient software packages for general simulation applications exist. It is difficult to predict the

quantity and quality of packages now in the planning stage.

The biggest disadvantage of the AD-10 vis a vis other leading contenders is that it is a fixed-point computer. This does not pose a problem in communicating with a host computer, since fixed/float conversion hardware will be included in the interface controller. The real problem lies in the need to scale the arithmetic operations in the arithmetic processor unit. This can be a time-consuming and frustrating task.

An additional potential disadvantage is the limited data word size. For real-time aerospace simulations, this should not be a major problem since larger words are generally required only for integration; and the integrator module employs 48-bit words. Furthermore, most host computers under consideration themselves employ 16-bit words.

#### Credibility of the Unit

The first prototype version of the AD-10 was introduced in the summer of 1976. Several other prototypes have been constructed since then, but none are in actual use. It is reasonable therefore to expect that some problems and bugs remain to be resolved. ECL integrated circuits are used extensively in the AD-10. These devices have an indifferent reputation for reliability - a price one pays for high speed.

Initially at least, the mean time between hardware failures of the AD-10 would not be exceptionally attractive. Although the AD-10 constitutes a departure from the normal product line of Applied Dynamics, sufficient electronic design capabilities exist in-house to assure that a reliable and satisfactory product will ultimately result.

### Cost

An adequately expanded version of the AD-10, fully capable of handling a wide range of aerospace problems including the helicopter benchmark problem, will cost approximately \$85,000. Some additional allowance must be made for the preparation of various software packages other than the assembler.

### 5.3 Denelcor, Inc. - HEP

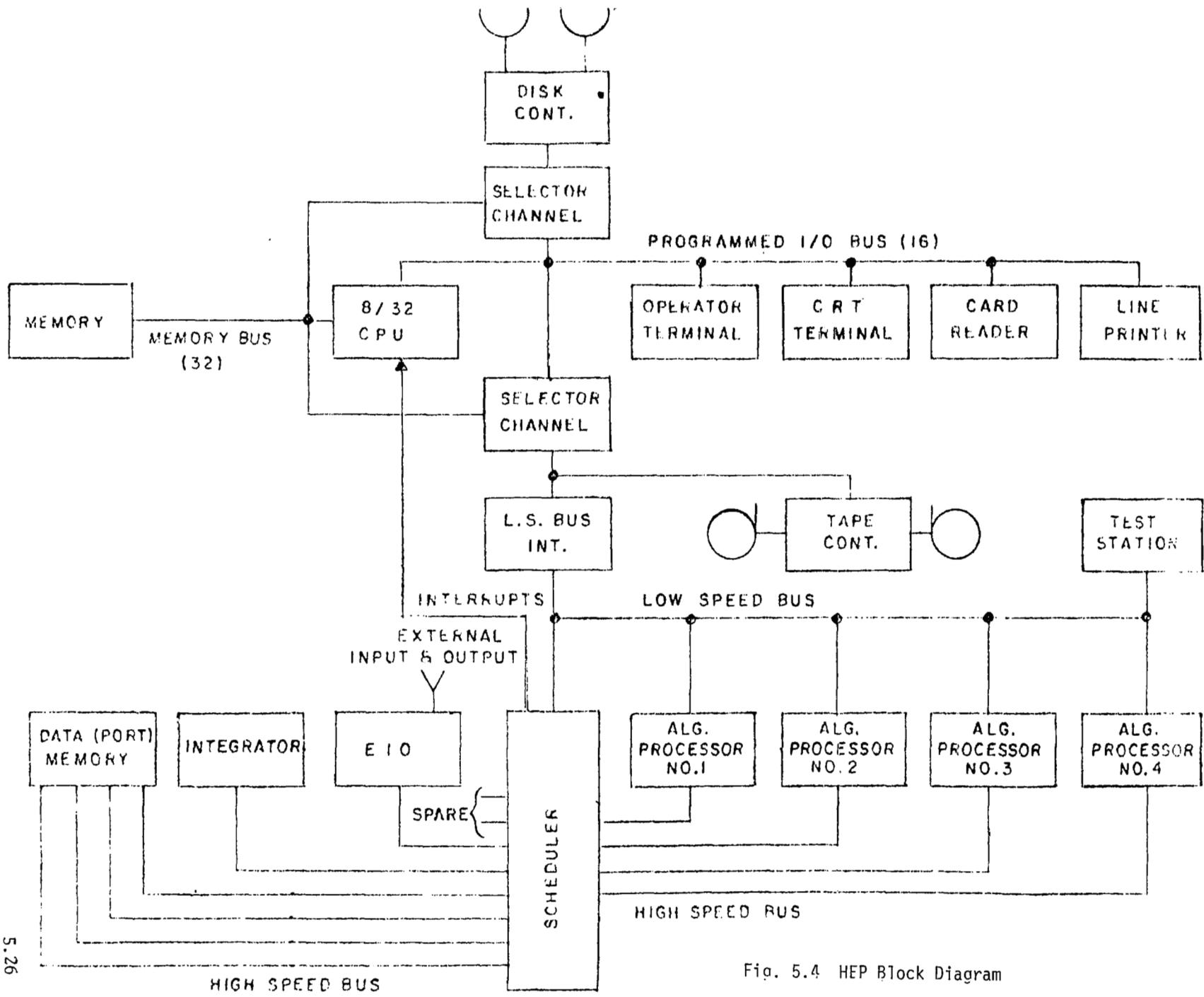
#### Overview

The heterogeneous element processor (HEP) is currently being developed for the Ballistic Research Laboratories. It is actually a full-fledged digital computer capable of large number-crunching tasks, but with an architecture that emphasizes parallelism. It is a MIMD (multiple instruction multiple data stream) machine as contrasted with the SIMD machines such as the STAR or ILLIAC IV. The method used in HEP to attain speed is to establish process execution in parallel on a number of processors. Synchronization among the processors is largely accomplished by hardware. A timing analysis indicates that the helicopter benchmark problem could be solved with a single processor in approximately 1.8 milliseconds. Using 4 processors operating simultaneously, this time would be cut to slightly under 0.5 millisecond. The cost of a four-processor system will probably be in excess of \$750,000. Software is currently being developed by Computer Sciences Corporation, Los Angeles. It is highly unlikely that any definitive system tests will be performed prior to the end of 1978.

#### Hardware Organization

The HEP system shown in Figure 5.4 consists of one control computer, a scheduler, and various computing and memory modules. The Algebraic Processor modules perform the high-speed computation, while the four Data Memory modules are used for high-speed data storage. The Integrator module implements automatic Runge-Kutta integration.

Three basic techniques are employed to achieve high speed. First, separate data and program memories are used, so that fetching an



5.26

Fig. 5.4 HEP Block Diagram

instruction can be performed in parallel with the execution of the previous instruction. Second, the processors of HEP are pipelined so that several instructions may be executed simultaneously. Third, the instruction words of the HEP processors are wide so that many actions can be specified in a single instruction.

Floating point data are in a 56-bit hexadecimal sign-magnitude format (48-bit fraction). The algebraic processes contain 512 registers of 64-bits each and are capable of supporting up to 64 multiple processes. Sophisticated queueing and scheduling techniques are employed to optimize data transfers.

#### Mode of Operation

If the HEP is used in a helicopter simulation, the entire computation would be performed in HEP. The host computer would be used principally as an interface to the communication lines and to control input/output devices. The entire program for the computation would be resident in HEP and would be carried out under its control.

#### Speed

Table 5.3 constitutes a timing estimate for the benchmark problem. If  $N$  represents the number of rotor blades and  $s$  is the number of finite elements per blade, the total time  $T$  (in microseconds) for a time frame when using a single processor is

$$T = 34.6 + 33.5 N + 65.0 Ns \quad (5.3)$$

If four parallel processors (which is the full expansion) are employed, the time required for a time frame is

$$T = 8.7 + 8.5 N + 16.2 Ns$$

TABLE 5.3

FRAME TIME OF THE DENELCOR HEP FOR  
THE BENCHMARK PROBLEM

	<u>One Processor μsec</u>	<u>Four Processors μsec</u>
Operations Independent of N and s		
131 Multiplications	13.1	3.3
95 Additions	9.5	2.4
6 Functions of One Variable	<u>12.0</u>	<u>3.0</u>
	34.6	8.7
Operations Proportional to N		
92 Multiplications	9.2N	2.3N
63 Additions	6.3N	1.6N
9 Functions of One Variable	<u>18.0N</u>	<u>4.5N</u>
	33.5N	8.5N
Operations Proportional to N and s		
30 Multiplications	30.0Ns	7.5Ns
13 Additions	13.0Ns	3.2Ns
2 Functions of One Variable	4.0Ns	1.0Ns
6 Functions of Two Variables	<u>18.0Ns</u>	<u>4.5Ns</u>
	65.0Ns	16.2Ns
Time for N x s = 25	1.83 millisec	0.46 millisec

N - number of rotor blades

s = finite elements per blade

For a helicopter with five rotor blades and five segments per blade, the frame times become 1.83 milliseconds and 0.46 millisecond respectively. It is of course difficult to anticipate in advance the amount of overhead that must be added to the above times, but 20% would appear to be a reasonable figure. Considerable time can be saved if the binary search for functions of two variables can be replaced by direct address calculations.

#### Programmability

As yet, no software packages have actually been written, and only general decisions have been made as to the overall software configuration. Denelcor has a contract with Computer Sciences Corporation, Los Angeles, to prepare a variety of packages required by the Ballistic Research Laboratories. No doubt a number of these packages would be useful for the simulation application. Still it would appear that an extensive in-house programming effort would be required if a HEP were acquired by NASA.

#### Principal Shortcomings

If actually realized and implemented as planned, the HEP would be more than adequate for all of NASA's simulation requirements.

#### Credibility of the Unit

HEP is essentially a one-of-its-kind system being designed and produced for a single customer. A preliminary prototype module was built and demonstrated. The system now under development profits from the experience with the prototype module, but it is considerably different in design and technology. ECL logic is being employed in HEP.

Past experience has shown this type of solid state device to be more prone to hardware failures and difficulties, but considerable progress has been made recently in increasing its reliability. It would appear that the Ballistic Research Laboratories' contract provides Denelcor with sufficient financial support to produce a working system. Only time will tell whether this system actually lives up to its specifications and whether adequate software will be available.

#### Cost

It is difficult to forecast the cost of a configuration suitable for simulation requirements. It would be reasonable however to expect the system to cost somewhere in the \$750,000 to \$1,000,000 range.

#### 5.4 W. W. Gaertner Research Inc. - G-471

##### Overview

The G-471 is an array of processors providing a high degree of multi-processing and pipelining. No versions of this unit have been built or are under construction. The proposed system would therefore be the first and only version. The proposed configuration appears however to be well within the state of the art and employs only off-the-shelf modules. Depending upon the number of processors used, frame times from 0.2 millisecond to 0.8 millisecond for the helicopter problem should be attainable. In addition to the cost of the host computer, which could be a PDP-11/70, the cost of the G-471 would be in the range of \$50,000 to \$200,000 depending upon the number of parallel processors acquired.

##### Hardware Organization

A block diagram of the proposed G-471 is shown in Figure 5.5. The control computer can be any standard general-purpose computer such as a PDP-11. It controls the operations of the processing elements and data-routing element arrays. The processing elements in the PE array are standard microcomputer boards each of which processes 16 or 32 bits in parallel. The local memory associated with each PE is expandable to at least 56 K bytes. This storage area can be assigned to data or program in any mix. PE's can readily be paralleled, with arrays ranging from 16 to 1024 PE's. Each PE can directly address up to 16 megabytes of semiconductor RAM central working storage. This working storage is partitioned into at least as many memory banks as there are PE's so as to permit parallel access. Each PE also has access to mass memory, typically a bank of discs.

TABLE 5.4

FRAME TIME OF THE W. W. GAERTNER G-471 FOR  
THE BENCHMARK PROBLEMS

	<u>One Processor μsec</u>	<u>Five Processors μsec</u>
Operations Independent of N and s		
131 Multiplications	35.4	7.1
95 Additions	17.1	3.4
6 Functions of One Variable	<u>3.4</u>	<u>0.7</u>
	55.9	11.2
Operations Proportional to N		
92 Multiplications	24.8N	5.0N
63 Additions	11.3N	2.3N
9 Functions of One Variable	<u>5.0N</u>	<u>1.0N</u>
	41.1N	8.3N
Operations Proportional to N and s		
30 Multiplications	8.1Ns	1.62Ns
13 Additions	2.3Ns	0.46Ns
2 Functions of One Variable	1.1Ns	0.22Ns
6 Functions of Two Variables	<u>21.06Ns</u>	<u>4.21Ns</u>
	32.6Ns	6.51Ns
Time for N x s =25	0.70 millisec	0.14 millisec
N = number of rotor blades	s = finite elements per blade	

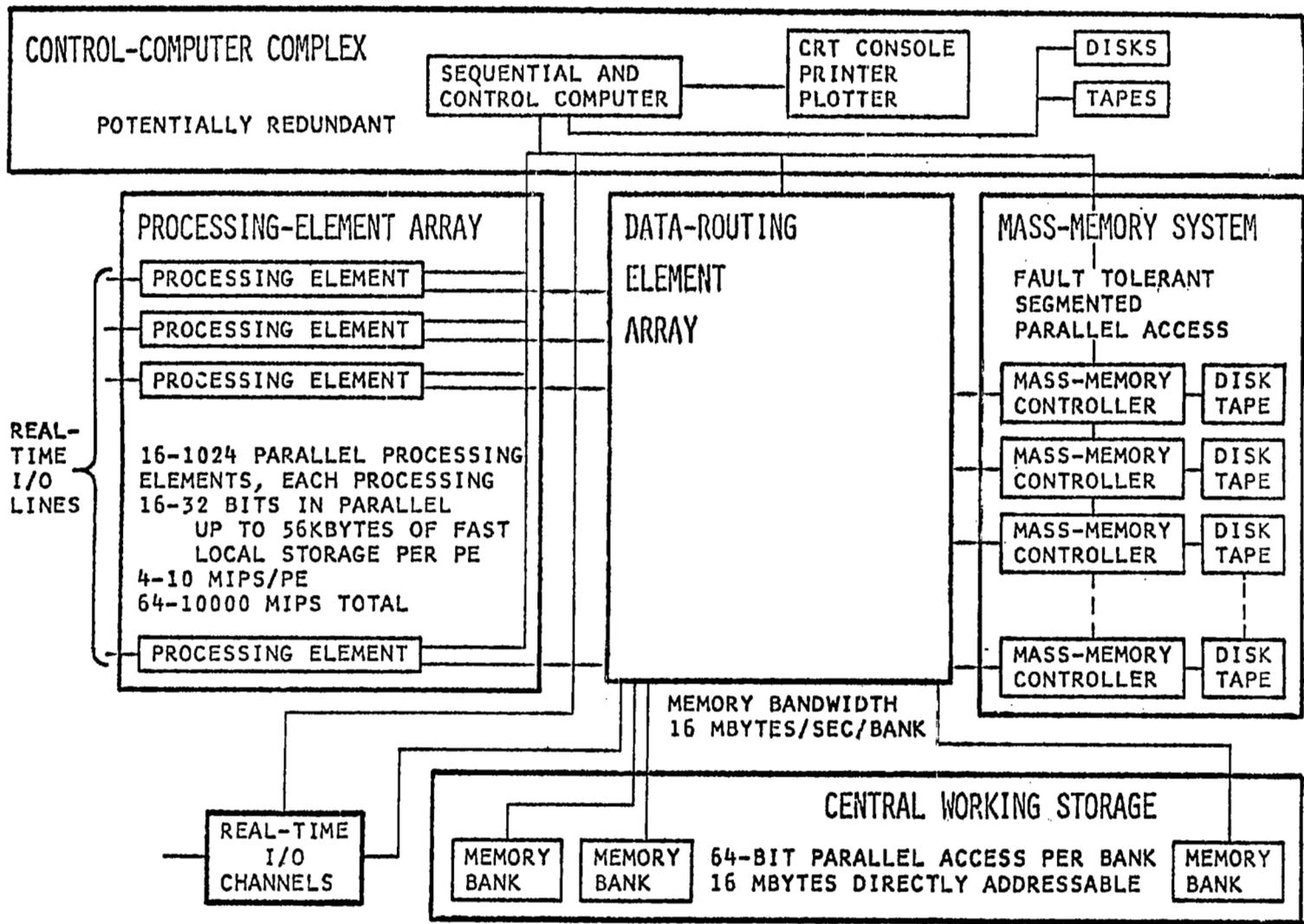


Fig. 5.5 G-471 Block Diagram

The data routing element array performs the communication functions among the PE's, the real-time I/O channels, the central working storage memory banks and the mass memory modules. It is basically a programmable cross-point switch whose switch settings are determined dynamically. An important feature of the design is the large amount of off-the-shelf hardware which is used.

#### Mode of Operation

The entire simulation program should be executed by the G-471 in order to profit from its high speed. This program would be read into the memories of the Processing Elements at the beginning of the computer run. For the helicopter simulation a simple Processing Element would be adequate. For increased speed a separate Processing Element might be dedicated to each rotor blade.

#### Speed

The G-471 operates on a clock which provides a 90 ns cycle time. The times which would be required for a single time frame of the benchmark problem are presented in Table 5.4. If five rotor blades with five finite elements each are employed, the total time  $T$  in microseconds required by a single processor is

$$T = 55.9 + 41.1 N + 17.9Ns \quad (5.5)$$

If five processors are employed, so that a separate processor is assigned to each of five rotor blades, the total time becomes

$$T = 11.2 + 8.3N + 3.59Ns \quad (5.6)$$

For a helicopter including five rotor blades with five finite elements per blade, the total time for a single processor system becomes 0.70 millisecond. The frame time for the same problem using five parallel processors is 0.14 millisecond. It is difficult to determine how much should be added to these figures for overhead and data transfer to and from the host computer.

#### Programmability

No software for the G-471 exists at the present time. W.W. Gaertner would however be willing to provide a standard assembler to facilitate the programming task. Clearly the optimal operation of an array of processors requires considerable special skills and a large investment in software. On the other hand, for the contemplated simulation applications nowhere near optimum operation would be required. The software problem is therefore not nearly as formidable as it would be in the case of applications requiring a large number of processors, each operating essentially independently.

#### Software Support

W. W. Gaertner proposes to provide a fully designed and implemented assembler to aid in the programming task. If a single processing element is employed, the programming would be quite straightforward, since no parallelism would be involved. The program would merely take advantage of the extremely rapid floating-point arithmetic capability of the processing element. Where a number of processing elements are used, some scheduling and queueing problems must be solved. This would place a premium on application programming skills.

### Application Support

The G-471 was proposed primarily with signal processing applications in mind. W. W. Gaertner has no direct experience with dynamic simulations. No support along those lines could therefore be expected from the vendor.

### Principal Shortcomings

If realized as proposed, the G-471 should meet all of the simulation requirements of NASA.

### Credibility of the Unit

The G-471 is strictly a proposal. No version of this unit has as yet been built nor are there any orders or contracts for it in existence.

### Cost

According to Dr. Gaertner a G-471 with a single Processing Element would cost \$50,000. For a system with six parallel processing elements the cost would be \$200,000. In this cost, is included a moderate amount of software development including the basic assembler.

## 5.5 Paragon Pacific, Inc. - SPURS

### Overview

The Special Purpose Helicopter Simulator (SPURS) is a special purpose hybrid computer, designed specifically for real time simulation of rotorcraft. However, in contrast with traditional analog and hybrid computer technology SPURS is hard-wired, using entirely integrated circuit technology. As a result the simulator is extremely compact in physical size since it lacks completely the patch bay which characterizes traditional analog computers. Its designers are engineers with a number of years of experience in helicopter simulation, both analog and digital, who see in the SPURS concept an economical and feasible solution to the real-time simulation problem. The parallel nature of the analog computation modules not only results in extremely high operating speeds, clearly compatible with real time, but in fact make possible a different approach to the solution of the aerodynamic load equations of the rotor blade. Rather than using finite element approximations, as is done with all the digital computer implementations, the analog integrators in SPURS make it possible to integrate continuously along the rotor blade.

The very advantages of compactness and low cost which characterize the SPURS concept, also indicate some of its limitations since re-programming can be an extremely difficult operation, involving either mechanical adjustment of large numbers of coefficient potentiometers, the installation of expensive digitally-set pots, digital units, or actual physical replacement of coefficient cards in the computer by the cards corresponding to another helicopter. A change of model from

helicopter to another class of vehicle may be completely impossible. The computer is readily capable of being interfaced with minicomputers of the type being considered, and its cost is moderate, probably in the vicinity of \$120,000 including the necessary software support packages. In contrast to some of the other candidate systems discussed in this chapter, the SPURS would come completely programmed to solve the helicopter equations of motion, with only coefficients and simplifications or changes left to the purchaser.

#### Hardware Organization

SPURS is mounted on a standard 19 inch rack and requires less than two feet of vertical space. It is constructed of circuit boards which plug into a mother board. Each circuit board is wired with a specific type of analog component, i.e., there is an amplifier board, a non-linear component board, a coefficient board, and so forth. The solution of the equations is done entirely by analog circuits with digital logic being present to sequence certain operations, as is discussed below. All components are permanently interconnected by means of wirewrap connections. A single SPURS box may contain 300-400 analog amplifiers and associated components, more than enough to simulate the helicopterequations of motion, including rotor blades with aeroelastic degrees of freedom.

Figure 5.6 illustrates the general hardware organization of SPURS, as required for solution of the aeroelastic rotor mathematical model. It consists of a high frequency analog section in which the mode shapes associated with particular elastic modes are stored and used as inputs to function generators. The function generators produce the aero-

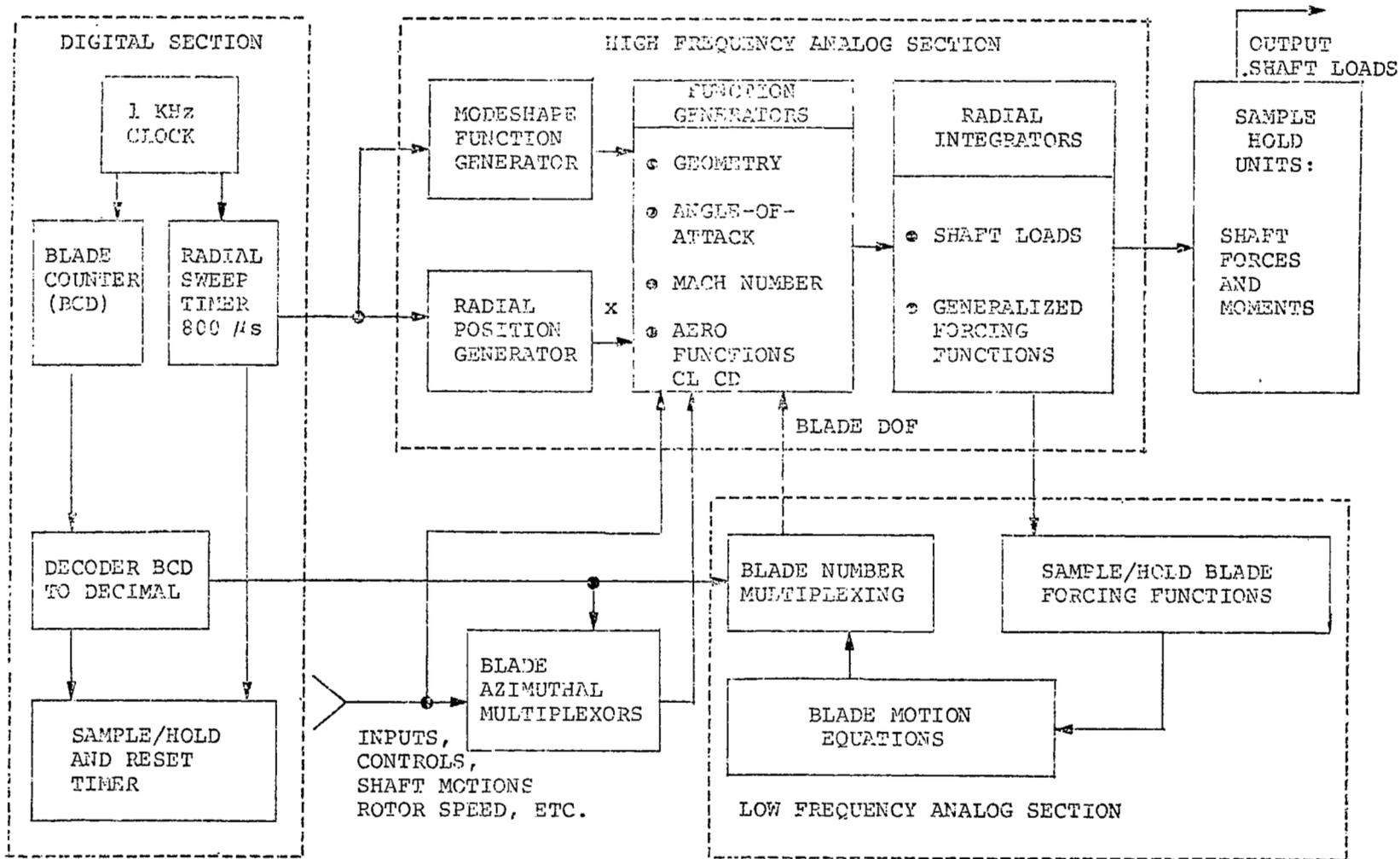


Figure 5.6- SPURS AEROELASTIC ROTOR SIMULATION

dynamic functions ( $C_L$  and  $C_D$ ) from angle of attack and Mach number at each radial position. Radial integrators are then used to compute the aerodynamic loads continuously as a function of radial position, from the hub to the tip at high repetitive rates. A digital control section is used to sequence the radial integrators from blade to blade. The remaining equations are solved continuously in a low frequency analog section.

The digital section is essentially an executive monitoring sequencer for the high speed analog section. The nonlinear equations associated with a rotor blade, are integrated once each 800 microseconds, with an additional 200 microseconds required for a "hold" needed to output the results and an additional 100 microsecond "reset". Thus, a five blade rotor simulation would require five milliseconds per "frame". Of course, the parallel nature of the analog elements in SPURS means that the same 800 microseconds are required regardless of the complexity of the equations or the number of bending modes included in the mathematical model. Furthermore, the equations of motion of the vehicle itself are integrated continuously and in real time.

The machine being manufactured for delivery to Fort Monmouth will interface with a cockpit through a digital host computer, on which the stability and control augmentation system, display generation, and other pilot related functions will be implemented.

#### Mode of Operation

Since the computer programming for SPURS is inherent in its wiring, its operation requires only the setting of coefficient values, and the programming of the host computer. Coefficient values can be set in a number of ways:

A. Coefficients on a coefficient board can be screwdriver adjusted. In view of the fact that any particular coefficient in the mathematical model may appear in numerous places throughout the simulation, Paragon Pacific makes available a computer program as part of the SPURS package which runs in the normal batch mode on any standard digital computer and calculates all the coefficient settings which are required for any particular change in helicopter parameters. The coefficient boards can then be adjusted and inserted into place at the beginning of the simulation.

B. Digitally set coefficients could take the place of the potentiometers provided in the standard model. Of course, such a provision would require the addition of the necessary logic for coefficient setting and it would significantly increase the cost of a unit, probably by at least \$20,000. A punched paper tape produced by the Paragon off-line program could then be used to set the digital coefficient units.

#### Speed

The question of speed is confusing with the hybrid computer when compared with the digital processors discussed earlier. While the specific SPURS under construction for the U.S. Army solves the rotor blade aerodynamic equations in one millisecond per blade, it is important to note that this time does not change when aeroelastic degrees of freedom are added to the rigid flapping and lagging modes discussed in the benchmark problem. The addition of three aeroelastic modes would probably double or triple the computation time estimates for the all-digital processors. Furthermore, the helicopter equations of motion are being integrated continuously. Hence, a five

millisecond frame time is in fact the maximum frame time to be expected, regardless of the degree of complexity to be included in the model. For example, the addition of stall aerodynamics, detailed representations of downwash or more complex tail rotor models would have no effect on the frame time.

#### Programmability

The SPURS comes completely programmed for the helicopter equation of motions. Where changes in the model are required, it is possible to insert coefficients into the equation which can then be set to 1 or 0, thus including or not including specific terms in a simulation. There is sufficient flexibility within the set of equations programmed in this manner to make it possible to use only coefficient settings to change among a wide variety of helicopter configurations. Clearly, changes such as number of rotor blades, mode of attachment of the blade at the hub, shape of the fuselage, and so on can be easily accommodated. It is not clear whether a vehicle such as a lift fan, or some other type of V/STOL can be accommodated by means of a SPURS box specifically designed for the simulation of rotorcraft. Paragon Pacific provides a digital support system which uses a modular stability derivative program (MOSTAB) to calculate all the necessary coefficients.

#### Application Support

Of all the companies considered, only Paragon Pacific has staff members who clearly understand the helicopter application, and hence could be counted on to provide significant amounts of application support. As indicated above, the computer would come fully wired for

the helicopter simulation. Should alternative Paragon computers be required for other vehicles, Paragon staff could be counted on to program the necessary equations in preparation for the construction of another hard-wired machine.

#### Principal Shortcomings

The SPURS is a special purpose computer, designed for simulation of the helicopter equations of motion in real time. Its lack of flexibility is its principal shortcoming.

#### Credibility of the Unit

As of the time of this writing, SPURS had not yet been delivered to Fort Monmouth. However, a computer with the same type of design features for simulating ship motion was demonstrated to the authors at Paragon Pacific headquarters in El Segundo, California, and performed impressively.

#### Cost

A SPURS unit completely programmed for real time helicopter simulation including aeroelastic degrees of freedom would cost approximately \$120,000. It is important to note that this price includes the digital support programs, and that no additional software investment would be necessary, except for the programming of the host computer and its interface with the cockpit.

## VI COMPARISONS AND TRADEOFFS

This chapter is devoted to a discussion of the relative advantages and disadvantages of the leading candidates for the simulator system. It is assumed that a peripheral processor will be connected to a small general-purpose digital computer such as the Digital Equipment Corporation PDP-11/70. The general purpose computer acting as a host will handle all communications with the cockpit hardware, as well as with a variety of terminals and other input/output equipment. In addition the host will perform a very small amount of the calculations required during each time frame of the simulation. These calculations will be limited to those blocks of the mathematical model which need to be revised frequently and which do not require high-speed computations. Virtually all the differential equations involved in the mathematical model will be handled by the peripheral processor in a way that requires only a single input vector at the beginning of the time frame and a single output vector at the end of the time frame.

Of the leading candidates described in the preceding chapter, the Floating Point Systems' AP-120B is available as an off-the-shelf item; the Applied Dynamics AD-10 has been available in prototype form but requires some minor additions for the present task; Paragon Pacific's SPURS has been constructed for a different application and would require some additional hardware development; the Denelcor HEP and the Gaertner G-471 only exist on paper at the present time. The fact that each of the five candidates is at a different point of development makes direct comparisons difficult. In the present chapter it has been assumed that

each of the five vendors will be able to produce a piece of hardware meeting its specifications, although it is realized that this is rarely the case in the design of novel and complex computer systems. From the point of view of hardware organization, the AP-120B and the AD-10 are fairly similar; the HEP and the G-471 likewise have many similarities. For this reason, a direct comparison is made below between the AP-120B and the AD-10. In Section 6.2, the advantages and disadvantages of choosing either the HEP or the G-471 rather than either the AP-120B or the AD-10 are considered. Finally in Section 6.3 the consequences of choosing the Paragon Pacific's SPURS hybrid system rather than any of the four digital processors is discussed.

## 6.1 Floating Point Systems AP-120B vs. Applied Dynamics AD-10

### Cost

The quoted prices of the AP-120B and AD-10 are very nearly the same. With the required memory expansions and interface units, both systems would run about \$80,000. However substantially more expensive and better-developed software packages are available for the AP-120B. Software development costs would therefore be higher for the AD-10.

### Speed

In real-time simulation applications the AD-10 is from 5-8 times faster. The speed advantage of the AD-10 is due to its faster cycle time, its more powerful arithmetic unit (which performs two additions and one multiplication simultaneously) as well as to the special hardware which is provided for memory mapping and binary search. For the benchmark problem, the AP-120B with an estimated frame time of approximately 3.5 milliseconds would be marginally acceptable. For other aerospace simulations, particularly those which require the evaluations of many functions of three or more variables during each frame, the AP-120B may be too slow. By contrast, the estimated AD-10 frame time for the helicopter benchmark problem was 0.6 millisecond, which provides considerable leeway for overhead and model growth.

### Accuracy

The AP-120B is a 38-bit floating point machine with a 28-bit mantissa. By contrast the AD-10 is a 16-bit fixed point machine. The AD-10 is therefore subject to a substantially larger round-off error during each arithmetic operation. Experience has shown however that in aerospace simulations, a 16-bit word is sufficient for all computations except the

execution of the integration algorithm; and ADI is providing an integration module with a 48-bit word. It can therefore be concluded that both systems are more than adequate in providing the overall accuracy sought in real-time simulations.

#### Program Size

The architecture of the AP-120B makes it impossible to expand the Program Memory beyond 4,000 instruction words. This places a ceiling on the complexity of the calculations which could be handled during a given time frame. The AD-10 does not have such a ceiling. For the helicopter benchmark problem, the program memory of the AP-120B is probably adequate. It would probably not be large enough for a number of other important simulation problems.

#### Flexibility and Suitability

The AD-10 is a more flexible device and has been specifically developed for simulation applications. It is more readily expandable and contains provision for the direct input of data from external communication lines. The design of the AP-120B is more or less frozen, and was directed to a signal processing rather than a simulation application. All information into and out of the AP-120B must go through the host computer, which may prove to be a significant bottleneck under certain circumstances, though not in the case of the helicopter benchmark problem.

#### Programmability

The AP-120B is far easier to program. Because of its fixed point data representation, the programming of the AD-10 poses significant scaling problems. The extent to which this is a damaging disadvantage

remains to be evaluated. Also because of its distributed program memory, the near-optimal programming of the AD-10 can be expected to be more difficult than that of the AP-120B.

#### Software Support

At the present time, the available software for the AP-120B is far superior to that of the AD-10. Of particular importance are the assembler and the simulator (emulator) packages of the AP-120B. It remains to be seen to what extent Applied Dynamics will be able to develop comparable software packages.

#### Application Support

Applied Dynamics is capable of giving significant and substantial support in application programming and in the planning of real-time simulations, having a long and impressive record in this field. Floating Point Systems has virtually no simulation experience and will therefore be able to give little if any application support.

#### Field Experience

Over 150 models of the AP-120B have been delivered to customers. An impressive mean time between failures (3800 hours) has gradually been achieved. Probably, the AP-120B will be considerably more reliable than the host computer and its peripherals. Although several versions of the AD-10 have been constructed, none have been used in the field. Moreover, the AD-10 employs solid-state circuitry which is inherently less reliable than that used in the AP-120B.

#### Summary

Both the AP-120B and the AD-10 seem adequate for the helicopter problem. For other and more complex problems, the AD-10 has a significant advantage. The AP-120B has a much more impressive track record to date.

## 6.2 AP-120B or AD-10 vs. Denelcor HEP or Gaertner G-471

### Cost

The AD-10 or AP-120B are less expensive. The cost for the G-471 would range from \$50,000 to \$200,000, depending upon the number of parallel processing elements. The Denelcor HEP would cost in excess of \$750,000. By contrast the AP-120B and the AD-10 both run about \$80,000.

### Speed

The G-471 and the HEP are substantially faster. This speed advantage is not evident for the benchmark problem since that problem is really too easy to take full advantage of the parallel processing capabilities of the HEP and the G-471. No doubt there are some simulation applications in which the HEP or the G-471 would have more impressive speed advantages.

### Accuracy

Both the HEP and the G-471 are capable of employing sufficiently wide data words to obviate any round-off error problems. This is also true of the AD-10 and the AP-120B.

### Program Size

Both the G-471 and the HEP are capable of handling sufficiently large programs. Only the AP-120B has a serious program memory size problem.

### Flexibility and Suitability

Both the HEP and the G-471 appear to be well-suited to simulation applications. The same is true for the AD-10 but not for the AP-120B.

### Programmability

The G-471 would appear to be relatively easy and straightforward to program, provided the simulation problem can be broken up into reasonably

independent segments. This is the case for the helicopter problem where each rotor blade calculation can be performed essentially independently. The HEP would require more challenging and difficult programming effort. Provided adequate assemblers are implemented, the programming of the AP-120B, the AD-10 and the HEP would be of approximately equal difficulty.

#### Software Support

No software exists at present for either the HEP or the G-471. Both of these candidates are therefore far behind the AP-120B and the AD-10 for which some useful packages are already available. Denelcor has already contracted for the development of HEP software packages. G-471 software would have to be developed from scratch.

#### Application Support

Applied Dynamics Inc. and Denelcor Inc. both have substantial simulation experience. Floating Point Systems and Gaertner Research have none.

#### Field Experience

Neither the HEP nor the G-471 have actually been realized as working hardware. By contrast there is a lot of field experience with the AP-120B and considerable prototype experience with the AD-10.

#### Summary

The HEP and the G-471 constitute more advanced designs with a greater degree of parallelism and flexibility. It is unlikely that their development will proceed sufficiently rapidly to make them suitable for acquisition in the near future.

### 6.3 Paragon Pacific SPURS vs. Digital Processors

#### Cost

SPURS is essentially a hardwired analog unit. Its hardware organization therefore includes the equivalent of applications programming for digital processors. The cost of approximately \$120,000 for the helicopter simulation is therefore probably comparable to that of the AP-120B or the AD-10.

#### Speed

Because it is a parallel analog machine, SPURS does not have an inherent speed limitation. The general approach can be expected to provide adequate speeds for all realistic simulations.

#### Accuracy

SPURS will probably function with adequate accuracy for all realistic simulations. It is not subject to truncation and round-off errors to the same extent as are digital processors.

#### Program Size

The size of the problem that can be handled with SPURS is limited by the actual hardware components which have been purchased and installed. This is of course not the case in digital processing.

#### Flexibility and Suitability

SPURS is probably ideal for a specific helicopter problem. An entirely different unit would have to be acquired for other kinds of aerospace simulations. Even with digitally-set potentiometers and some reprogrammability, SPURS is probably not sufficiently flexible for the general facility. By contrast, digital processors are more awkward and less suitable for a specific simulation such as the helicopter

problem, but they possess the capability of being rapidly adapted to virtually all other simulation requirements.

#### Software Support

The SPURS system comes essentially completely programmed, so that no software problem exists. Digital processors require a considerable amount of special software.

#### Application Support

Paragon Pacific is expert in helicopter simulation. The support they could provide for that problem is therefore far superior to that available from the other vendors. On the other hand, Paragon Pacific has virtually no experience with other kinds of aerospace simulations, so that they would not be as good as Applied Dynamics or Denelcor in that area.

#### Field Experience

Some versions of SPURS have been constructed and delivered to the U.S. Army. They will probably function as reliably as any of the digital processors under consideration.

#### Summary

SPURS is the ideal solution for a specific helicopter simulation, but it is not useful for any other simulation task.

## VII CONCLUSIONS

The following conclusions are based upon the analyses presented in Chapters V and VI. It should be emphasized that the comparisons are based primarily upon the helicopter benchmark problem. No detailed consideration was given to computational requirements other than for the solution of the differential equations characterizing the dynamic processes in helicopters. However, it is probable that this is the most demanding application of the proposed simulator.

1. A host computer of moderate size supported by a peripheral processor is capable of meeting the simulation requirements.
2. The five leading candidates for peripheral processors are the Floating Point System Inc. AP-120B, Applied Dynamics Inc. AD-10, Denelcor, Inc. HEP, W. W. Gaertner Research Inc. G-471 and Paragon Pacific SPURS. The time required by each of these processors for all of the computations needed to solve the differential equations for a single time frame is

	AP-120B	AD-10	HEP	G-471	SPURS
Frame Time (millisec.)	3.5	0.60	0.46	0.14	5.0

All of these are fast enough for the helicopter benchmark problem. With the exception of the AP-120B, all are so fast that the time required by the peripheral processor will be substantially overshadowed by the time required by the host computer for data transfers, communication, etc.

3. It follows that considerable care must be devoted to the selection of the host computer and to the architecture of the overall system. The Digital Equipment Corporation PDP-11/70, possibly supported by a peripheral PDP-11/45 may be adequate for the task. There are however a number of other general-purpose digital computers which may provide more speed at approximately the same cost (\$150,000-\$200,000).
4. The cost for an adequately expanded peripheral processor including some basic software packages is

AP-120B	AD-10	HEP	G-471	SPURS
\$80,000	\$85,000	\$750,000	\$200,000	\$120,000

With the exception of HEP, all of the peripheral processors constitute relatively moderate additions to the overall cost of the integrated system.

5. The five leading candidates are at radical states of readiness. As of February 1977, only the AP-120B was available as an off-the-shelf item. The AD-10 hardware as well as rudimentary software packages is expected to be ready during the second half of 1977. HEP can be expected to be available for preliminary testing no earlier than the end of 1978. The G-471 and SPURS would be manufactured upon receipt of an order. Thus, if it is necessary to make a procurement immediately the AP-120B is the only choice.

6. Minimum required software support for all of the digital processors includes an assembler, a simulator (emulator), a subroutine library, and diagnostic packages. All vendors have indicated that they would furnish an assembler and some subroutine and diagnostic packages. Only the AP-120B has a simulator available. The preparation and debugging of utility programs and application programs can be expected to be a large though not overwhelming task.

## VIII RECOMMENDATIONS

On the basis of this study, the following course of action is recommended:

- 1) Decide to accept the concept of employing a general-purpose digital computer as a host, supported by a peripheral digital processor.
- 2) Eliminate all analog and hybrid computing devices from further consideration. However, Paragon Pacific's SPURS should be recognized as a promising back-up possibility. Should the digital peripheral processors fail to provide adequate performance for specific problems, a SPURS could be acquired to meet such a specific requirement.
- 3) A detailed and systematic study should immediately be undertaken to prepare the specifications for the host computer and the communication links to the cockpit stations. This study should be conducted in-house by a team intimately familiar with the mode of operation of the Simulation Laboratory and the requirements of the users of the Laboratory. This study should lead to the selection of a host computer or a host computer complex with an optimum combination of computing speed, input/output capability, and cost.
- 4) Further detailed application studies should be undertaken before procuring the peripheral processor. These studies would be directed along three avenues:
  - i) Detailed study of the applicability of the Floating Point Systems Inc. AP-120B. Of particular importance are further considerations of the consequences of the limitation of the size of the program memory, interfacing problems, and general programming difficulties.

As part of this study a major portion of the helicopter benchmark problem should be programmed in AP-120B assembly language and implemented on a large general-purpose computer, using the AP-120B simulator program. This would provide a very convincing measure of the suitability of this peripheral processor.

- ii) A more detailed study of the applicability of the Applied Dynamics AD-10 should be made. This study should monitor the continued hardware development of the AD-10, particularly the implementation of the integration module, as well as the evolution of software packages. Applied Dynamics Inc. has indicated that it will shortly commence the development of a FORTRAN-based assembler, but the basic structure of this program is still open to discussion. ADI should also be encouraged to provide a simulator (emulator) program along the lines of that currently available for the AP-120B. A major portion of the study would be directed toward a careful evaluation of the implications of the fixed-point nature of the AD-10 particularly as concerns scaling. More general programming difficulty should also be analyzed in more detail so as to determine whether the programming of the AD-10 is too difficult on the long run to warrant the procurement of this processor.
- iii) A continuing study should be made of alternative candidates for the peripheral digital processor. Denelcor's HEP is currently under construction for the U.S. Army; a version of Paragon Pacific's SPURS is currently being tested at Ft. Monmouth; and a number of new entries into the array processor market are eminent. The increased availability and reliability of ECL

circuitry makes it likely that very high-speed arrays of processors will become available. The implication of all of these developments upon NASA/AMES simulation requirements and plans should be studied on a continuing basis, with periodic reports and presentations.

## APPENDIX

### List of Organizations and Individuals Contacted

#### Suppliers and Vendors

Applied Dynamics, Inc. - R. Howe, G. Graber, D. Chandler, E. Fadden,  
E. Gilbert  
Control Data Corporation (MAP III) - A.C. Champlin, D. Dawkins  
CSP, Inc.  
Datawest Corporation  
Denelcor, Inc. - M. C. Gilliland, R. Lord, B. J. Smith  
Electronic Associates, Inc. - A. Rubin, P. Landauer  
Floating Point Systems, Inc. - J. Sherfey, F. Krueger, R. Norin  
IBM, Los Angeles (Federal Systems Division) - E. Peirola  
IBM, (Owego, New York) - J. Caldwell  
MacNeal-Schwendler Corporation - R. MacNeal  
Magnavox Research Laboratories - T. Wetkowski  
Paragon Pacific, Inc. - John H. Hoffman  
Probe Systems, Inc. - Carroll Keilers  
W. W. Gaertner Research, Inc. - W. Gaertner

#### Organizations Involved in Simulation Activities

Ft. Eustis - Edward Austin  
Information Sciences Institute - T. O. Ellis, Randy Cole  
Informatic, JPL - J. Dennis  
Jet Propulsion Laboratories - Dr. Gerald Burnham  
Lockheed Corporation (Burbank) - H. Hara, D. Kawamoto  
MacNeal-Schwendler Corporation - R. MacNeal  
NASA/Langley Research Center - Dr. R. Bowles, J. Houck, J. Copeland  
National Science Foundation - Dr. H. Rigas, Dr. M. Wozny, J. Lehman  
Systems Control, Inc. - Dr. A. Phatak, Dr. E. Hall  
TRW - Dr. J. Maloney  
U.S. Army Electronics Command, Ft. Monmouth, N.J. - Dr. N. Shupe, R. Pribyl  
U.S. Army Material Command - A. Saucier  
USC Aerospace Engineering Department - Dr. R. Bucy